Libre Software as an Innovation Enabler in India Experiences of a Bangalorian software SME

Katja Henttonen

VTT Technical Research Centre of Finland
Oulu, Finland

www.vtt.fi
katja.henttonen@vtt.fi

Abstract. Free/Libre and open source software (FLOSS) has been advocated for its presumed capacity to support native software industries in developing countries. It is said to create new spaces for exploration and to lower entry barriers to mature software markets, for example. However, little empirical research has been conducted concerning FLOSS business in a developing country setting and, thus, there is not much evidence to support or refute these claims. This paper presents a business case study conducted in India, a country branded as a 'software powerhouse' of the developing world. The findings show how FLOSS has opened up significant opportunities for the case company, especially in terms of improving its innovative capability and upgrading in the software value chain. On the other hand, they also highlight some challenges to FLOSS involvement that rise specifically from the Indian context.

Keywords. open source, innovation, India, free software, software business

1 Introduction

Free/Libre and open source software (FLOSS) has been widely advocated [e.g. 1-4] as a way to promote endogenous software innovation in developing countries. The developmental opportunities created by the FLOSS phenomenon have been noticed both by international development institutions (e.g. World Bank and UNDP) and many of the developing countries themselves [1,3,4]. However, despite the enthusiasm, there remains very little empirical research on how developing country companies could successfully integrate FLOSS efforts into their internal innovative activities. Studies on commercially-motivated FLOSS in the US and Europe abound, but the results may not be directly applicable to the diverse innovation environments in the global South. This paper presents some key results of a qualitative case study [5] conducted in India, the country with the most well-known software industry in the developing world. The aim is to understand FLOSS-created opportunities and challenges from the viewpoint of an indigenous software SME.

The focus of the study is on the impacts of FLOSS on the innovativeness and profitability of the case company. Herein, innovativeness means the ability to create and implement new ideas which generate commercial value [cf. 6]. This can entail improvements to products, internal operations or a mix of markets. The study concerns modest incremental innovations, which an SME can generate on a regular basis.

The rest of the paper is structured as follows. The second chapter is divided into two sections: the first summarizes theoretical concepts underlying the study and the second one briefly introduces the current debate on whether and how Indian primary software sector could benefit from FLOSS.[1] The third chapter describes the research approach and methods employed in this study, and also very briefly introduces the case company. The fourth chapter presents the actual case study results; it is organized in three sections reflecting three different approaches to open innovation (more on these below). The fifth chapter discusses the meaning of some findings for further research. Conclusions close the paper.

2 Background

2.1 FLOSS as open innovation: three archetypes

This study builds on the Chesbrough's [7] Open innovation theory, which describes the recent tendency of companies to 'open up' their innovation processes. In open innovation, not all good ideas need to be developed internally, and not all ideas should necessarily be further developed within firm's boundaries [8]. Chesbrough and Crowther [9, cf. 10] distinguish two archetypes of open innovation: inbound and outbound. In the case of inbound open innovation, companies monitor the surrounding environment of the firm to find technology and knowledge to complement in-house R&D. In the case of outbound open innovation, companies are looking for external organizations to take internally developed technology into new markets. An additional approach to openness is an interactive value co-creation in strategic partnerships [11, cf. 10] Here, the focus is on innovating together rather than on bringing resources over company borders (inside or outside) [8].

From a perspective of a private company, FLOSS involvement becomes open innovation when it is combined with a sustainable business model [12]. The aforementioned 'subtypes' of open innovation can be used to categorize how primary software companies engage with FLOSS [5, cf. 12,13]. In inbound open innovation, a company sources free-of-charge intellectual property (IP) from FLOSS communities and uses it

[1] The focus is on introducing the points put forwards in the development literature; the discourse is somewhat different in the FLOSS business literature. For the comparison of discussions in the two disciplines, please see [5]

to produce commercial software products or services. Typically, the main goal is to save own R&D expenses and/or achieve faster time to market[2]. The outbound open innovation entails what West and Gallagher [12] call “open source spin-out”: a company brings internally developed IP into FLOSS domain. It may aim to to create demand for associated commercial offerings or advance strategic goals such as standards creation, for example. OSS communities can also be platforms for open value co-creation where diverse stakeholders join forces to achieve a common R&D goal and pooled contributions are made available to all [cf. 12].

2.2 FLOSS-based innovation in the Indian context

While Indian software exports have grown exponentially over the past two decades [15,16], many observers have pointed out that the industry's innovative capability has remained relatively low [15,17,18]. The vast majority of Indian software exports consists of low-value-adding off-shoring services such as maintenance of legacy systems [15,17,18]. Due to barriers such as heavy financial constraints, 'late-comer disadvantage' and geographical distance from key customers, many Indian software entrepreneurs struggle to upgrade in the software value chain [15,17]. Meanwhile, 'FLOSS debate' is getting heated: academics and policy makers are arguing [e.g.4,19-22] on whether FLOSS could help some Indian software companies, especially SMEs, to increase innovativeness, add more value and capture more returns.

The proponents point out that sourcing technology from FLOSS communities (i.e. inbound open innovation) saves R&D time and costs and can thereby help Indian companies to overcome financial constraints and 'catch-up' to older players on the global software markets [3,4,23]. Another key argument relates to inter-organizational learning through gradually deepening FLOSS participation (in open co-creation). Unlike off-shoring parent companies, who often have a strong incentive to prevent knowledge spill-overs, FLOSS communities are very motivated to share knowledge across organizational and geographical boundaries [24,25]. This is said to offer valuable learning opportunities to Indian and other Southern companies [2,3,22]. Interestingly, the possible benefits of outbound open innovation has not been discussed much in the development literature, perhaps reflecting a tacit assumption that relevant IP and technical knowledge flows 'from the West to the Rest' rather than vice versa.

Some critics have argued that any competitive advantage derived from FLOSS-enabled cost and time savings is mitigated by GPL-like licensing terms [19,26]. As these licenses make it difficult to sell mass-distributed packaged software, they are said to deprive Southern software companies from the opportunity to benefit from the 'economies of repetition' [19]. Others have pointed out that 'price parity' with pirated software is shirking the markets for FLOSS in the South [21,27]. It also widely acknowledged that the cultural and linguistic barriers may hinder learning trough parti-

[2] This does not necessarily equal to a 'parasite approach': a company may motivate external innovation, e.g. by financially sponsoring FLOSS development [5, cf. 13,14].

cipative process in FLOSS communities [20,28].There are also significant differences between FLOSS communities on how they draw the boundaries of peripheral participation: some are highly inclusive, while others welcome only very advanced programmers [28,29]. Further, open co-creation and outbound open innovation both require significant investments in non-(directly) revenue generating activities [13] and because Indian companies typically face heavier financial constraints than their Western counterparts, affordability can become a major problem [20]. Launching an own FLOSS project is considered particularly costly and challenging human resource wise [30-32].

Somewhat surprisingly, despite the lively debate, empirical studies on FLOSS activities of primary software companies in India are almost non-existent. Some authors [e.g. 19] even dismiss the subject by saying that FLOSS plays no role in the Indian software industry. However, an international survey [33-35] indicates that, while commercially-motivated FLOSS involvement seems to be a relatively weak phenomenon in India (e.g. in comparison to Europe or Brazil), many small FLOSS companies are still 'out there' and FLOSS experience is also highly appreciated by recruiters in more 'mainstream' software companies. The survey [34,35] also suggests that most Indian companies limit themselves to inbound open innovation as far as FLOSS is concerned. Outbound open innovation seems particularly rare, only three Indian companies were found to author their own FLOSS projects [33]. Mahammodan and De [36] also analysed FLOSS reuse by six proprietary software producers in India. While these organizations reportedly attained significant cost savings by using FLOSS components as 'black box', their developers often lacked sufficient time or skills to even read the source code, leave alone contribute back.

3 Research approach and methodology

The paper presents results from a single case study conducted in a company called Mahiti Infotech Private Limited[3] (in short 'Mahiti') which is headquartered in Bangalore and employs 70-90 people. The company employs a customized product development model [37]: it develops 'semi-finished' products, often co-creatively with FLOSS communities, and later adds value by customizing them to the needs of individual end-clients. The tailored products go to market either as bespoke software or through the application service provision (ASP) model. Technical consulting provides additional revenue streams.

While planning to conduct more case studies in the future, the author believes that findings from this one case study alone may be valuable for the research community. Especially so, because, to the knowledge of the author, no previous academic study has aimed to 'give a say' to FLOSS entrepreneurs in India. Further, even though the case cannot be argued to be perfectly 'revelatory' nor 'exemplifying' in a strict sense

[3] Researchers have argued both for and against disclosing the organization's name in case studies, see [47] for an overview. In this study, the company directors were given a choice and they selected recognition over anonymity.

[cf. 38], there are certainly many interesting characteristics to it. For example, despite its relatively small size, Mahiti has an extremely visible role in the Indian FLOSS scene. It can also be regarded as a notable example of an SME which has successfully used FLOSS strategically in order to upgrade in the software value chain. The case company also integrates elements from all the three archetypes of open innovation, thereby allowing to analyse outbound/inbound open innovation and open co-creation within the same organization.

The primary method of data collection was semi-structured interviews of the case company personnel. Three directors, the company's chief executive officer (CEO), chief technical officer (CTO) and marketing director were interviewed along with few senior developers. Two other sources of evidence, documentation (e.g websites and mailing list archives) and unobtrusive observation (mostly of employee interaction on FLOSS related IRC channels) were used to collaborate and augment evidence collected in the interviews. In order to cross-check data further [cf.39,40], some questions were also made to representatives of partner organizations. Most interviews were recorded and transcribed; in few cases, it was necessary to rely on note taking instead. A qualitative method called Template Analysis [41] was employed to thematically analyse the interview transcripts and, to a much smaller extent, some documentary evidence. In short, this means that a coding template was developed iteratively whilst the analytical process moved forwards. The final template served as a basis for interpreting the data and writing up the findings. In addition to the thematic coding, some aspects of the Value Network Analysis [42] approach were used. The role of this method was complementary and it is not elaborated herein.

This study aims to confirm to the criteria that Guba [43,44] suggests for qualitative research: credibility (a parallel of internal validity), dependability (a parallel of reliability) and transferability (a parallel of external validity). To improve *credibility*, the study relies on several data sources and two different analysis methods as explained earlier [cf. 40]. The results report has also been shown to key informants for confirmation [cf.39,45]. To ensure *dependability*, complete records have been kept of the collected raw data (a case study database) so that other researchers can check them per request [cf, 22,50]. As for *transferability*, the results from a single case study are not generalisable to other situations, but they can still contribute to the understanding of the target phenomena and thereby provide valuable leads for future research [40,46]. Further, a longer research report available online [5] provides additional contextual information which can help others to make judgements on the transferability of the findings to other settings [cf. 38].

4 Case study results

4.1 Experiences in inbound open innovation

In order to save costs, Mahiti intensively encourages the use of FLOSS code and components in all of their software projects. One of the founders gauged that an average Indian software company pays approximately 15% of their profits back in licensing fees, an expense they avoided. However, the cost savings and their profitability implications varied a great deal in practice as illustrated by two recent customer projects (see Fig.1). In the first case, the company only needed to make minor modifications to an existing FLOSS product, but could still charge a 'premium price', higher than that of all proprietary software vendors participating in the bid. This is because the FLOSS product in question met well with the needs of the customer as such and Mahiti could offer the fastest lead-time. 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, but could still charge a much lower price. The project was not profitable alone, but was still worth doing because the extension developed in this project was expected to be resold to several other customers over time.

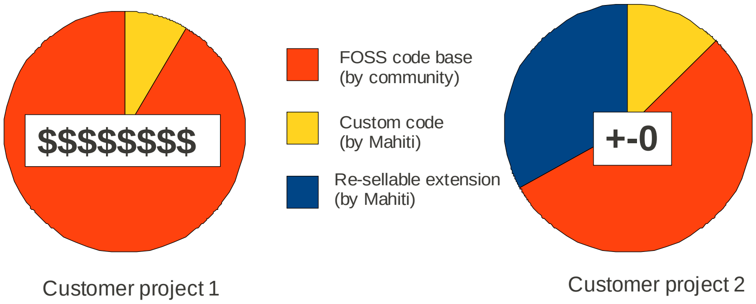

Fig. 1. Proportions of FLOSS and 'own' code in two projects [5]

FLOSS also brings cost savings to customers and, according to Mahiti's experience, this is helping to expand bespoke software markets in India and other developing countries. Interestingly, unlike most Indian software SMEs [15,18], Mahiti has highly diversified export markets with customers in countries such as Mongolia, the Bahamas, Brazil and Tajikistan. They believed this is partially because FLOSS based solutions are more affordable to Southern customizers, though it is obvious that many other factors are also at play. Nevertheless, it is noteworthy that, while the ease of piracy diminishes the cost advantage of FLOSS on the realm of mass software, the impact is not the same on the bespoke software markets. Pirated mass products can be customized to a certain extent (e.g. Microsoft Excel with macros), but such possibilities are very limited.

FLOSS licensing did not cause any fundamental changes to the company's revenue models[4]: instead of tailoring proprietary software packages, they were customizing FLOSS solutions. The latter allowed them to add-more value in-house, thanks to the low 'purchase price' and unlimited customization options. Profiting from the 'economies of repetition' through closed-source product development was seen infeasible due to financial constraints and highly mature markets: “basically, the curve to recover the funds is very high and this kind of [business] model is not viable for a company like ours”. To the question of whether FLOSS licensing terms limit profit-making, the CEO replied:

– “Yes, 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 wrote. But I cannot 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.”

When asked about the main difficulties in FLOSS reuse, directors pointed to difficulties in finding recruits with any previous experience on FLOSS technologies and development practices. This was seen as stemming from the tendency of local engineering education to ignore the skills needs of FLOSS companies and from the relatively small number of volunteer FLOSS developers [cf. 33] in the country. There was a recognition of some recent positive developments on the field of education. However, while some FLOSS technologies were slowly making their way to the engineering curricula, general code reuse skills were reportedly not. Consequently, the vast majority of new recruits were totally unfamiliar with the whole concept code reuse, only vaguely associating it with 'cut and paste'. They had to be taught 'by hand' which tended to bring up training expenses. As a strategy to address the skills gap, the company has started to offer free-of-charge lectures on FLOSS skills to local engineering colleges.

For the case company, another concern is that, as the vast majority of FLOSS projects originate from the global North [cf. 48], they do not always address regional needs as well as locally created software could. For example, the directors pointed out that there are practically no FLOSS applications addressing non-urban development needs in India, such as monitoring the quality of water or coordinating rural healthcare. “All of these are possible with FLOSS, but there are very few projects moving despite a huge demand”, the CTO said. He added that, while many FLOSS applications are relatively easy to localize in terms of language or metric systems, there are also more fundamental differences in software requirements between countries and regions. Referring to the cultural diversity within India, he continued: “this is a vast

[4] For more information on FLOSS licensing issues in the case context, please see [5].

country and on the way from Bombay to Delhi the requirements change also... so no matter how much FLOSS there is in a market, it is not enough.”.

4.2 Experiences in outbound open innovation

Mahiti is one of the very few Indian software companies [cf. 33] to author its own FLOSS development project. Recently launched OurBank (www.ourbank.in) is a micro-finance software community which has attracted dozens of volunteer developers, mostly Indian engineering students, and NGOs from as far as Brazil have contributed localization effort. Based on their positive experience, Mahiti's directors are convinced that it is feasible for a resource-constraint SME to launch its own FLOSS community. Success on this arena was seen to depend on “energy and passion” as well as certain key capabilities (e.g. social networking skills) rather than large monetary investments. On the other hand, the CEO did admit that capturing returns from FLOSS spin-outs can be difficult: “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 [FLOSS customization projects].”

The mentioned profit-making challenges exist despite some institutional donations (e.g. from EuropeAid) towards the development of OurBank. However, most difficulties were believed to relate to the incubation phase. In the long run, Mahiti plans to step aside from leading the community and become just one of the many contributors. Such partial 'hand-off' was seen necessary so that the community can “evolve naturally”. Time will show how the transition will work out in practise.

Apart from the spin-out described above, Mahiti has a longer history in doing releases which could be called ‘spin-offs’. Whenever they have a piece of 'surplus' source code, which has reached the end of it's life cycle, they put it freely available on SourceForge or similar OSS platform. Because nothing is invested in community building or even making a website, the cost of open-sourcing is very low in this case. The company reported concrete and significant benefits once the IP got 'new life' in FLOSS domain. For example, they once open sourced a very 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.

4.3 Experiences in open value co-creation

Mahiti also plays a globally important role in the development of some FLOSS products such as the Plone content management system. When asked about business gains from strategic FLOSS participation, the global marketing benefits were typically mentioned first. The company does not need to engage in conventional marketing, directors said, because “FLOSS gives us complete visibility”. Being listed as a partner on the Plone website was alone considered to be a major advantage. Further visibility resulted from employees' contributions, which they were always advised to do under the name of the company, and from co-organizing Plone conferences. However, it was not only about having one's name visible but, more importantly, about being seen as a 'shaper' of the technology: “They [customers] come to us because they see us as people who vision the [Plone] product and not only as people having [third-party] expertise on it”, explains the marketing director. In addition, FLOSS communities are specialized social networks were 'word-of-mouth' travels quickly. Recommendations from other FLOSS community members brought in many customers. To exemplify such discussions, a UK-based member 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”.

Somewhat expectedly, another group of reported benefits related to inter-organizational learning. The employee training at Mahiti is closely integrated with FLOSS participation. New employees started by following discussions on FLOSS forums and they were encouraged to gradually deepen their participation, very much in line with the classic 'onion model' [49] frequently stated in FLOSS research. The interviewed employees seemed genuinely enthusiastic about this training method. One said that while engineering education had only taught her to complete specific tasks, FLOSS participation had taught her to find solutions independently. From the management viewpoint, there are cost advantages because new employees are coached free-of-charge[5] by external experts. Some drawbacks were also mentioned, for example, FLOSS project administrators did not always explain why they rejected a contribution, which obviously constrained what an employee could learn from the experience.

As to other forms of inter-organizational learning, the company had benefited from adopting process innovations from FLOSS communities. For example, FLOSS involvement had prompted the company to adopt and improve the practice of end-user co-development. As a result, intense collaboration with domestic end-customers, who paid below-market prices in return of participating in R&D and beta-testing, had become a key part of their innovation process. Further, as a result of their FLOSS activities, the company has become geared towards writing well-commented and highly modular source code which is easy to reuse internally. They have even introduced an 'internal source forge', a repository where developers search for reusable source code developed in previous customer projects. These new development practices have enabled the case company to move away from one-time-project development towards developing 'semi-packages' out of reusable modules. In this way, FLOSS had clearly become a booster rather than a barrier to the 'economies of repetition' discussed earlier.

[5] Alternatively, the coaching can be understood as a social return from investments which the company makes to foster its relationship with FLOSS communities [cf. 13]. One developer said that Mahiti's 'good reputation' in communities helped her to get assistance.

The challenges discussed in the context of inbound open innovation also have an impact on open co-creation. In addition, directors acknowledged there are economic barriers to making FLOSS contributions. However, Mahiti has found several ways to keep expenses reasonable. Making minor contributions like bug fixes is integrated into employee training as explained earlier. The company also intermediates contributions made by others, for example, it facilitated local Myanmarian refugees to translate Plone into Burmese and put their contribution online. Or, as in the Hecker's [50] “sell-it-free-it model”, large FLOSS contributions often consist of source code that has already been sold to one or more customers. The later model is not only an issue of affordability though; co-operation with end-customers was also seen crucial for ensuring the quality of the contribution. The CTO explains: “You cannot release something [to a FLOSS community] and expect miracles, unless you have tested it and the only way to test a product is to test with a customer...once it's a stable product only then the masses [of FLOSS users] can use it”. Reportedly, most of the company's customers do not mind a contractual clause saying that the source code developed for them might be open source later.

Interestingly, developers said that they had not experienced any language barriers to FLOSS participation. Tertiary education had left them with a decent command of English and, besides, they felt that only technical terms are needed to talk on FLOSS related IRC channels. This is not an ethnographic study and it was not possible to detect how more 'subtle' cultural or linguistic issues may shape their identity building as FLOSS developers and effect their sense of belonging to a FLOSS community. On the surface, however, the employees seemed to feel sincerely proud of being well-recognized and respected members of the FLOSS communities where they contributed. For example, they very positively recalled that Joel Burton from the Plone Foundation had visited Mahiti and socialized with them. This was understood to be evidence that their participation is highly appreciated. “If we worked with Microsoft, Bill Gates would not come to chat with us”, compared one.

5 Discussion

This paper does not aim to advocate Mahiti's experiences as a success model, which Indian software SMEs in masses could imitate. Firstly, it is appreciated that the study has succeeded to identify more opportunities than challenges. Despite cross-checking information from different sources (including non-company ones), the study still essentially relies on what the informants decided to share. Most people prefer to talk about their successes rather than their failures and the informants were supposedly not free from this common human tendency. Secondly, the case company seems to possess unique capabilities and also has a different market position than most Indian software SMEs. To exemplify the latter, Mahiti serves direct end-customers, over half of which are non-profit organizations. As such customers often agree to open-sourcing the code, which they have already paid for, the company can benefit from 'the sell-it-free-it' model. The scene is supposedly very different for most Indian SMEs, which do subcontracting work for multinational ICT companies.

The paper has 'scanned' several opportunities and challenges faced by the case company and none of these could be discussed in great depth. However, the author hopes that the paper has helped to highlight the wide range of perspectives, which one should take into consideration when discussing FLOSS business in India, or possibly other Southern contexts. For example, some development writers [e.g. 19] argue that FLOSS business models are 'less profitable' without discussing what are the likely alternatives for software companies in that particular country/region. Or, on the other side of the debate, the 'endogenous' nature of FLOSS is often strongly advocated [e.g. 1,4] without discussing the challenges that Southern organizations face when trying to launch their own FLOSS projects.

Most prior work on FLOSS-enabled learning, especially in the development context, focuses on technological knowledge transfer [e.g.2,25,29]. However, this study points to significant benefits from learning new development practices on customer co-development and code reuse. The study also suggests that FLOSS can have mixed impacts on the costs of employee training. These are both interesting subjects for further research, especially considering that the low level of code reuse (often below 5%) and high training expenses are often mentioned among key factors hindering profitability of Indian software SMEs [15,17,18]. Other topic, which deserves more attention, is the potential ability of FLOSS to expand low-cost markets for bespoke software in the South. The strong emphasis, which the interviewees placed on the global marketing benefits of FLOSS participation, is also noteworthy. Very hypothetically, this could related to the cost of international marketing (e.g. adverts in international magazines) being proportionally higher than the cost of R&D labour (i.e. FLOSS participation) for Indian companies.

From the viewpoint of the Open Innovation theory, Mahiti's experiences in uploading 'surplus' source code to the Internet are particularly interesting. Their habit strongly reflects one of the Chesbrough's [7] main “ethos”: one should never 'sit' on the surplus intellectual property. The case study hints that SourgeForge-like platforms might provide a low-cost route for releasing IP which is no longer creating value inhouse. If the released IP creates value elsewhere, there is a chance to claim a portion of that value. While getting theoretical support from Open innovation researchers [e.g. 51], this idea conflicts with many prior studies [13,30,32], which suggest that any commercially-motivated FLOSS release should be supported by significant investments in marketing and infrastructure building.

6 Conclusion

The study illustrates how FLOSS can blur the boundary between software vendor and third-party service provider, thereby opening up new opportunities for companies who lack resources to develop own products from 'scratch'. FLOSS co-creation has helped the case company to develop 'vendor-like' in-depth expertise and build an image as a co-creator of certain technologies. Due to the availability of source code and the absence of licensing fees, they can also add more value to FLOSS products than a non-vendor can typically add on proprietary products. In some cases, FLOSS releases have even helped to open up routes to new markets. Meanwhile, the case company continues to face many challenges such as the poor availability of new recruits with FLOSS competences in India. More research is needed to understand how the findings may apply beyond the single case setting and whether FLOSS has any potential to transform the Indian software sector at large.

7 Acknowledgements

I want to thank Mr. Pasi Pussinen, a research scientist at VTT, who gave some excellent feedback on a draft of this paper and all the informants, who spent their valuable time being interviewed. As to financing, this work has been supported by the research project ITEI-VTT, which is co-funded by VTT and Finnish National Technology Agency (Tekes).

References

1. Dravis P. Open source software. Perspectives for development. Global information and communication technologies department, the World Bank, Washington (2003)
2. Tapia, A. and Maldonado E. An ICT Skills Cascade: Government-Mandated Open Source Policy as a Potential Driver for ICT Skills Transfer. Information Technologies and International Development, 5(2),31–51 (2009)
3. Weerawarana, S. and Weeratunge J. Open Source in Developing Countries. SIDA, Stockholm (2004)
4. Wong K. Free/open source software: government policy, UNDP Asia Pacific Development Information Program in co-operation with Elsevier, New Delhi. (2004)
5. Henttonen K. Open source as an innovation enabler: A Case Study of an Indian Software SME. Dissertation, The University of Manchester. Institute for Development Policy and Management. (2011). http:/opensource.erve.vtt.fi/publications/henttonendissertation.pdf
6. Subramaniam M. and Youndt S. The influence of intellectual capital on the type of innovative capabilities. Academy of Management Journal , 48(3), pp. 450–463 (2005)
7. Chesbrough, H. Open innovation: the new imperative for creating and profiting from technology. Harvard University Press, Boston (2003)
8. Koskela K., Koivumäki T, Näkki P. Art of openness. In M. Pikkarainen, W. Codenie, N. Boucart and J. Heredia (eds.) The Art of Software Innovation. Eight Practice Areas to Inspire your Business. To be published by Springer (2011)
9. Chesbrough, H. and Crowther A. Beyond high tech: early adopters of open innovation in other industries. R&D Management, 36, 229–36 (2006)
10. Gassmann, O. and Enkel, E. Towards a theory of Open innovation: three core process archetypes. In the proceedings of the R&D Management Conference, Sesimbra, Portugal (2004).
11. Piller, F. & Ihl C. Open Innovation with Customers – Foundations, Competences and International Trends. Trend Study within the BMBF Project ‘International Monitoring’, RWTH Aachen University, Aachen (2009)

12. West J. and Gallagher S. 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 University Press, Oxford, pp. 82-106 (2006)
13. Goldman G. and Gabriel R. Innovation happens elsewhere. Open source as business strategy. Elsivier, San Fransisco. (2005)
14. Dahlander, L. and Magnusson M. Relationships between open source software companies and communities: Observations from Nordic firms, Research Policy, 34(4), 481-493 (2005)
15. Arora, A. The Indian software industry and its prospects, in J. Bhagwati, and C. Calomiris (eds.) Sustaining India's growth miracle, Columbia University Press, New York, pp. 166-215 (2008)
16. Athreye S. 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, Oxford University Press, pp. 7-14 (2005)
17. D'Costa A. Export Growth and Path Dependence: Locking Innovations, Software Industry, Science, Technology and Society, 7(1), pp. 51-81 (2002).
18. Nirjar, A. and Tylecote, A. 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 (2005).
19. Debroy, B. and Morris, J. Open to development: Open-Source software and economic development. International Policy Network, London (2004).
20. O'Donnell C. A case for Indian outsourcing: open source interests in IT jobs. First Monday, 9(11) (2004)
21. Sharma A. and Adkins R. OSS in India. In Dibona C., Cooper D and Stone M. (eds) Open sources: the Continuing Evolution. O'Reilly, Sebastobol (2005)
22. Suman A. and Bhardwaj K. Open Source Software and Growth of Linux: The Indian Perspective , DESIDOC Bulletin of Information Technology , 23(6), 9-16 (2003)
23. May, C. The FLOSS alternative: TRIPs, non-proprietary software and development. Knowledge, Technology, and Policy 18 (4), 142–163, (2006)
24. Krogh G., Spaeth S. and Lakhani K. Community, joining, and specialization in open source software innovation: a case study. Research Policy, 32 (7), 1217–1230 (2003)
25. Staring K. and TitleStad O. Development as a Free Software: Extending Commons Based Peer Production to the South. In the proceedings of the Twenty Ninth International Conference on Information Systems (ICIS 2008), Paris. (2008)
26. Reddy, B. and Evans, D. Government Preferences for Promoting Open-Source Software: A Solution in Search of a Problem. Social Science Research Network. (2002)
27. Heeks, R. Free and Open Source Software: A Blind Alley for Developing Countries?, IDPM Development Informatics Briefing Paper, Institute of Development Policy and Management. The University of Manchester. (2005)
28. Vaden T. and Vainio N. Free and Open Source Software Strategies for Sustainable Information Society. In O. Hietanen (ed.) University partnerships for international development: Finnish development knowledge, Finland Futures Research Centre, Turku (2005)
29. Wernberg-Tougaard C., Schmitz P. and Herning K. and Gøtze J.(Evaluating Open Source in Government: Methodological Considerations in strategizing the Use of Open Source in the Public Sector. In M. Lytras and A. Naeve, (eds.) Open source for knowledge and learning management: Strategies beyond tools. Idea Group Publishing, London (2007)
30. Henttonen, K. and Matinlassi, M. Contributing to Eclipse - a case study. Proceedings of the Software Engineering 2007 Conference (SE2007). Hamburg, Germany (2007)

31. Järvensivu, J., Mikkonen, T. Forging A Community? Not: Experiences On Establishing An Open Source Project. In B. Russo, E. Damiani, S. Hissam, B. Lundell, G. Succi (eds.) Open Source Development Communities and Quality, IFIP Working Group 2.13 on Open Source Software Systems (OSS2008), Milano, Italy (2008).
32. West, J. and O'Mahony, S. Contrasting Community Building in Sponsored and Community Founded Open Source Projects. In the Proceedings of the 38th Annual Hawaii International Conference on System Science, IEEE Computer Society, Los Alamitos (2005)
33. UNU-MERIT Free/Libre and Open Source Software: Worldwide Impact Study. D31: Track 1 International Report. Skills Study. Maastricht : United Nations University. (2007)
34. UNU-MERIT Free/Libre and Open Source Software: Worldwide Impact Study. D7: Track 1 Survey Report - India. Skills Study. Maastricht : United Nations University. (2007)
35. UNU-MERIT Free/Libre and Open Source Software: Worldwide Impact Study. D7: Track 2 Survey Report - India. Software Study. Maastricht : United Nations University (2007)
36. Madanmohan, T. and De. R. Open source reuse in commercial firms. IEEE software, 21(6), 62-69. (2004)
37. Codenie W. Pikkarainen M., Boucart N, Deleu J. Software innovation in different companies. In M. Pikkarainen, W. Codenie, N. Boucart and J. Heredia (eds.) The Art of Software Innovation. Eight Practice Areas to Inspire your Business. To be published by Springer. (2011)
38. Bryman A. Social Research Methods. Third Edition. Oxford University Press, New York (2008)
39. Chetty, S. The case study method for research in small and medium sized firms', International Small Business Journal 15(1), 73–85 (1996)
40. Yin R. Case Study Research: Design and Methods. Fourth Edition. Sage Publications, California (2009)
41. King, N. Template analysis. G. Symon, and C. Cassell (eds.) In Qualitative Methods and Analysis in Organizational Research: A Practical Guide. Sage Publications, California, pp. 118-134. (1998)
42. Allee, V. Value Network Analysis and Value Conversion of Tangible and Intangible Assets. Journal of Intellectual Capital 9(1), pp. 5-24. (2008)
43. Guba, E. Criteria for assessing the trustworthiness of naturalistic inquiries , Educational Technology Research and Development , 29(2), pp. 75-91. (1981).
44. Guba, E.G. and Lincoln, Y.S. Competing paradigms in qualitative research, In N.Denzin and Y. Lincoln (eds.) Handbook of qualitative research, London: Sage Publications, pp. 163—194 (1994)
45. Rowley J. Using case studies in research, Management Research News, 25(1), pp.16 – 27 (2002)
46. Flyvbjerg, B. Five Misunderstandings About Case-Study Research, Qualitative Inquiry, 12(2), pp. 219-245. (2006)
47. Walsham, G. Doing interpretive research, European Journal of Information Systems, 15(3), pp. 320—330 (2006)
48. Robles, G. and Gonzalez-Barahona, J.M. Geographic location of developers at SourceForge. In the Proceedings of the 2006 International Workshop on Mining Software Repositories, Shanghai, China, pp. 144-150, (2006)
49. Ye, Y. and Kishida, K. Toward an understanding of the motivation of open source software developers. The proceedings of the 25th International Conference on Software Engineering (ICSE'03). (2003)
50. Hecker, Setting up shop: the business of Open-Source software. IEEE Software, 16(1), 45–51 (1999).

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