

Citation: G. Gaughan, B. Fitzgerald, M. Shaikh (2009) An examination of the use of open source software processes as a global software development solution for commercial software engineering, in *Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications*, Greece, pp. 20-27.

## **An examination of the use of Open Source Software Processes as a global software development solution for commercial software engineering.**

Gary Gaughan [1], Brian Fitzgerald [1], Maha Shaikh [2]

*Lero The Irish Software Engineering research Centre [1], London School of Economics [2]*

*{Gary.Gaughan, Brian.Fitzgerald}@lero.ie [1] M.I.Shaikh@lse.ac.uk [2]*

### **Abstract**

*This paper outlines the phenomenon of Inner Source software development and places it in the context of existing open source literature. Our study includes an analysis of multiple case studies of Inner Source in use in large scale global software development companies. The lessons learned from these case studies help us to contrast traditional open source principles with Inner Source principles, and we then gather these lessons, to create our preliminary framework, in order to make sense of when and how firms can adopt Inner Source. Our framework helps to make sense of the practical issues of adopting and managing Inner Source. We have highlighted the emerging trends in the Inner Source phenomenon and surrounding areas. Awareness of this may be of great benefit to researchers in the area and industrial practitioners.*

### **1. Introduction**

In this paper we outline the principles of Inner Source extracted from our initial examination of emerging case studies and surrounding literature. Responding to the research challenges posed by Herbsleb [1] we aim to contribute to the call to make better sense of when and how to orchestrate global software development. In particular our motivation has been, a number of large-scale multinational organizations that have taken the practices of Open Source in house in a number of new initiatives and are reporting success. The success of Open Source beyond software has been an interest in both industry and academia for some time. We aim to characterize this phenomenon to enable us to research the topic in greater depth. Our detailed review of literature to date and in particular our focus on multiple case studies enable us to formulate a preliminary framework to help firms make sense of the emerging Inner Source culture,

and whether it could be an appropriate choice for industry in their global development scenario. The starting point for Inner Source is an understanding of open source and its many benefits and principles. The benefits of open source [2] are varied and relative to each organization. More and more we see large organizations focus heavily on open source [3]. We can see this focus not only by the amount of large corporate investment directed into open source projects<sup>1</sup>, (FLOSS Report, 2006) but also by the adoption of open source methods and practices behind corporate closed doors [4].

Open Source is seen by many as a successful example of large scale global software development (GSD) and as such, the open source development model is attracting considerable attention as organizations seek to emulate open source success in traditional development projects, through initiatives variously labelled as ‘inner source’, ‘corporate source’ [3], ‘community source’, or ‘iSource’. It has been argued that open source communities can be considered in some cases to be virtual organizations [5],[6]. If we consider this assertion to be true then the possibility of being able to transfer the success and benefits of open source to corporate organizations becomes more real.

Inner source, as we use the term here, is the leveraging of open source software development methods within a corporate environment. A variant of this term is progressive open source (and within it Inner Source) [7]. Inner Source is usually employed by companies to capitalize on the success that certain open source projects have enjoyed. However, there are some differences between open source and closed source development and their respective development communities. In traditional software development

---

<sup>1</sup> “Firms have invested an estimated Euro 1.2 billion in developing FLOSS software that is made freely available. Such firms represent in total at least 565 000 jobs and Euro 263 billion in annual revenue.” (FLOSS Report, 2006)

users and developers are typically located in separate departments and locations. This can lead to employees being unaware of development innovations and projects, and all too frequently to little mutual respect or voluntary interaction. The user-developer relationship has typically been quite different in open source. While early open source developers were users of actual products, as OSS has developed the situation has changed somewhat. In the absence of a traditional software development company, users need to become more intimately involved in the development process, as technical staff cannot simply send a checklist of requirements to the vendor to ascertain if needs will be met. It is a widely held belief that deploying open source can lead to a sense of shared adventure which is not a common scenario in the proprietary software arena [2]. The transfer of these lessons, where appropriate, to conventional in-house software development is a critical activity and is of much interest to these researchers and we are eager to leverage the lessons learned here for the industrial community.

We see that just as the principles of open source [2] have helped in understanding, defining and implementing open source, one of the most beneficial places to start is with the contrast between open source principles and inner source principles.

This paper will outline a number of case studies and exploratory interviews to give practical grounding in what initiatives are ongoing in the area. While the case studies have a lot of similarities in that they are for the most part positive experiences of Inner Source, this is probably due to firstly, a low adoption rate in a new emerging technology; and secondly, perhaps an unwillingness to publicize and to a lesser extent discuss the failures of initiatives within companies. We then set about examining two particular fundamental areas of Inner Source managing and organizing Inner Source and the principles of Inner Source. We examine the important issues of managing and organizing an inner source solution which as we note require significant cultural change. Against this background we then present Inner Source principles gathered from the case studies and other relevant literature. The initial understanding and definition of Inner Source is examined and refined throughout the sections of this document.

## 2. Methodology

We designed the study in two key stages, the first interpretivist case study stage gave us an overview of current Inner source solutions and the state of the art. The second stage involves exploratory key interviews to

confirm or deny our beliefs and to investigate queries that will arise. Given the nature of this new and rising topic and the limited number of implementations, we feel an approach to the research study such as this was appropriate. Our interpretive [8],[9],[10] study includes seven case studies [11] of large global companies, Bell Labs, Hewlett Packard, IBM, DTE Energy, Nokia, Microsoft, and Philips. Case study research is appropriate when a phenomenon is complex, when a holistic, in-depth investigation is needed, and when a phenomenon cannot be studied outside the context in which it occurs [12],[13],[11]. Three out of the seven case study material is literature based. We reviewed both academic and industrial literature to confirm which companies were using Inner Source ideas. A thorough review of the literature and our search brought a number of companies to our attention but we wanted to focus on only those firms that had implemented Inner Source for two years or more years and where there was in-depth literature available on their experience. For a number of reasons most of the literature to date has focused on somewhat successful cases of Inner Source use. This then required us to study the documents and literature very carefully to reach an objective and balanced understanding.

Our understanding of the literature guided our interview-based exploration into the last four cases. This paper is part of an ongoing study, and the process of interviewing will continue over the next six to eight months. Our initial interviews were focused around a set of semi-structured questions, and these questions emerged from our literature based study and our framework.

### 2.1. Literature Study

Seminal work on each of the cases was identified and references followed up in academic work. Industrial reports and news articles on the same companies were scrutinized. Our data collection and data analysis stages were iterative. We built criteria for analyzing literature based on a number of categories like our definition of Inner Source, length of project duration in the company, domain area of the projects, and the size of the company (our focus is on multinational companies which tend to large in size). This list evolved over the early part of our data collection and analysis. New categories emerged over our study and we finally decided on three main cases based on our criteria and the recognition that only some cases had been academically analyzed with any detail. This provided us with a natural way to end our search and focus on three main cases.

Our analysis of the documents allowed us to immerse ourselves into the context of the company and we began to notice certain characteristics and issues for each case. There were similarities in all three cases yet each was also distinctive enough to allow our framework of principles to have wider appeal and use.

## 2.2. Interviews

At the time of writing this paper we conducted 7 key interviews, 3 of which were with developers and 4 with middle managers. The interviews were informal and were conducted with an interview guide so as to be productive in exploring the areas that were considered either to be of great interest or remained ambiguous. A typical interview lasted between thirty minutes and one hour. The number of interview was not set from the beginning rather a number of questions were raised in our initial study and once these questions were adequately addressed and confirmed by interview, it was deemed satisfactory to allow us to analyse the data and address the next actions to take in our research. Interviews were recorded and notes taken at the time in response to predesigned interview guide. This was later transcribed and discussed for answers, themes and ideas. The interviews have no affiliations with the researchers.

## 3. Case Studies

We examined seven different unique implementations of Inner source. The majority of implementations examined involve large scale multinational corporations. The singular exception is DTE energy, an American energy company based in Detroit Michigan. The case studies are primarily based on literature review and supplemented by informal telephone or face to face interviews where ambiguity remained after the initial literature review. The preliminary interviews in some cases with public relations PR personnel were removed as no new information came to light. The remaining interviews developers or middle management. Developers (3) PR (8) middle managers (4).

Each of the cases were examined on how they classified (not the term Inner Source) their particular adoption of Open Source methods. Differences here obviously exist but of the cases included all fall under our umbrella term Inner Source. Other studies were examined and discounted as a result of examination. We also examined the motivational factors for attempting to implement change, the expected results

versus actual, the nature of change required, problems encountered and unexpected results.

The initial list of terms was acquired on the early examination of the case studies; this was a complete list and included every factor reported under the 5 exploratory headings (motivation, problems, benefits, project type, and common theme). These headings were chosen as many of the cases tended to focus on these particular issues. Once the categories were determined a second data collection pass was used to gather the data. A third pass confirmed the findings in case of error. The figures below were chosen, as the decision was made to present only the data that was deemed by the authors, to offer some insight or surprising factors. Some of the initial findings that proved insightful have been presented below in figure 1 and 2. These exercises in examining the nature of Inner Source helped guide us in the extraction of principles and characterization of Inner Source in the following sections.

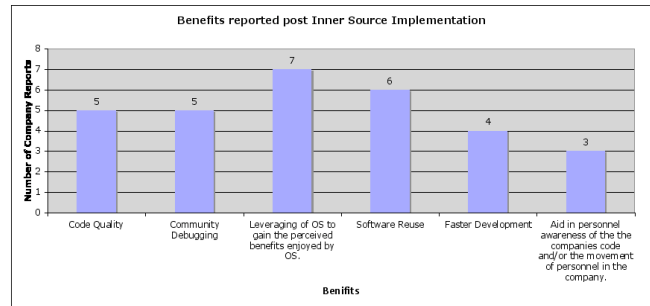


Figure 1.

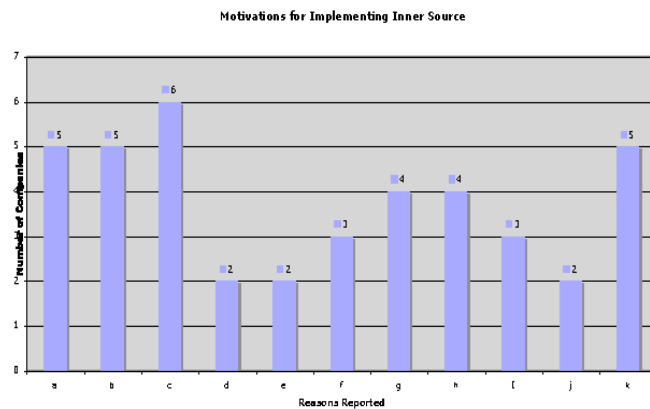


Figure 2.

The table above outlines the motivations of firms implementing Inner Source, the y axis illustrates the number of firms reporting the individual motivating factor and the x axis illustrates the motivating factors (a to k) involved. That is:

- a Global Software Development
- b Software Reuse
- c Overcome problems that OSS is reported to have achieved.
- d Maintenance
- e Quality
- f Many Eyeballs
- g Community Building
- h Open Discussion
- I Modular Design
- j Greater Agility
- k Decrease Costs

#### 4. Discussion

In the previous section we examined multiple cases studies. Each case was discussed in terms of its background, how Inner Source was applied within each company, the implications and problems that arose as a rest of the implementation, the key results drawn from the case study and an analysis of each case study. In certain cases the project chosen was typically an emerging area of interest in the industry with a dedicated domain expert as the champion of the project. As the project grew in maturity it would attract more attention and benefit from the snowball effect where attention would gain more developer participation and achieve a yet more mature stage.

The cases examined helped establish ongoing implementations of Inner Source and the issues surrounding such implementations. We now understand the:

1. Differences between Inner Source and traditional software development.
2. Differences between Inner Source and open source.
3. Perceived benefits of Inner Source usage.
4. Actual benefits of Inner Source usage.
5. Problems with implementing inner source solutions and some of the possible factors causing these problems.
6. Principles on Inner Source.
7. The cultural changes required for a successful Inner Source solution.

This leads us to make some arguments concerning firstly, how inner source is managed and organized (Section 5) and secondly, a step closer to characterizing Inner Source (Section 6).

#### 5. Managing Organizational Change and Inner Source

In this section we look at how traditional organizational and management structures have changed in order to accommodate and benefit from Inner Source adoption. The case studies indicate that there are two clear advantages of adopting Inner Source use within companies stemming from the increased visibility offered by open source principles, but also a number of problems but they are all related to the idea of visibility afforded by the open source aspect of Inner Source.

*Better quality code through visibility:* Visibility is an important issue to consider in Inner Source. As traditional projects move from away from traditional style development, where the code and communication is usually shared only within the project group and where face to face communication is common, to Inner Source style development visibility of the developer's work and textual communication increases significantly. The issues surrounding this may include an increase in quality of work as a result of many people being able to give helpful feedback but it also takes more time and effort for people to communicate in this environment.

*Pride in their work through greater visibility:* In HP, through increased visibility, the employees were conscious of posting to the Inner Source community as their messages were going out to a wider community. Managers felt that the use of the Inner Source solution would lead to an increase in quality due to the open nature of the Inner Source solution. People were insecure about posting their work for a company wide audience but eventually felt a lot of pride associated with contributing quality work [14].

*Security issues surrounding visibility:* However, as visibility increases so too must the companies' security efforts, especially in the case where various contractors are employed within different areas of the company. Where in the traditional development environment the access of people to code is localized within company departments and easier to manage, in contrast the very openness sought after in the use of inner source makes security and access control a lot more difficult. In HP digital badges were used to control access and vendor access to digital badges was something new that had to be managed as a result of the use of Inner Source. This creates an additional issue to be dealt with when bringing in a new vendor, as not only must the new vendor have a physical ID but also a digital badge for Inner Source access [14]. This creates a certain level of trust within the company that the code is safe but when using external vendors the perception is that they have

no control over what security the vendor is using once the vendor is given access.

*Fear of job loss through visibility:* Some of the fears encountered as a result in the increase in openness include fear of granting access to external access of the code to vendors and also by allowing other groups to see what is happening, the risk is perceived to be higher, of someone seeing an opportunity to downsize the department and take over the work. The other fear among developers is that by everyone in the company potentially having greater access to the inner workings of the company any employee that leaves permanently may bring an increased skill set and knowledge to competitors and as a result threaten jobs in the company.

*Privacy and knowledge retention issues related to visibility:* There is also the argument that in an environment where there exists a centralized repository where a lot of information is freely available on a company wide basis and if free to traverse and browse, that this environment could make people reluctant to use digital media for some forms of communication. This was certainly the case in HP where developers rejected the suggestion of the use of video equipment to record idea generation and brainstorming in a meeting direct because of the likelihood that the media would be made freely available in a company wide system where the media could live for a significant period of time.

*Increased visibility leads to easier workplace monitoring:* Introduction of an Inner Source system that is perceived to be open and a controlling factor by management causes serious undermining of developers. If Inner Source is to be used effectively and to promote innovation then introduction of an Inner Source system can aid decentralized and geographically dispersed development teams to work together on a project. The openness of the Inner Source solution deployed in HP caused developers to be a lot more careful about the quality of their work and comments they were contributing. The openness of the system also caused some users to become very concerned over job security. It is clear that management and organizational structure must change and adapt to the use of Inner Source as must the users of the system.

Our initial work had outlined the background difference between traditional software and Inner Source software development and the corresponding problems arising as a result of a change in environment. While some of these problems are a direct result of implementing inner source not all problems have a net negative effect. For example the problem of code forking, depending on the situation, is probably better than the previous situation where code

was developed in parallel with little information about other ongoing project development. At least in the case of code forking the developers are aware of other code existing and possible solutions to existing bugs.

There are of course cases of cross-over between both types of development for example Garcia [15] argues that most of the characteristics found in traditional development are found in open source development, and in particular Capiluppi and Michlmayr [16] argue that many open source projects never migrate from centralized development to distributed development which are commonly held polar opposites between traditional development and open source development [2].

Another example of this cross-over is when companies attempt to build a community around a corporate product [17] where companies which released developed code to a public community much in the same way Nokia release code to the Maemo community.

## 6. Characterization of Inner Source

In this section we characterize Inner Source (see Table 1). Our analysis of the case studies and subsequent interviews, allowed us to organize the main arguments and issues emerging over the use of Inner Source ideas in companies. We identified these to fall within two broad categories of *product* and *process* (see Table 1). Open source can be usefully analyzed through ideas of product and process [18] and thus this categorization is also, to a degree, useful for Inner Source studies.

**Table .** Characterizing Inner Source

Inner Source Characteristics		
<b>P r o d u c t</b>	<i>Domain area</i>	A common area to choose for Inner Source projects was that of research or emerging technologies.
	<i>Economic issues</i>	Costing of code produced in the new method was not valued correctly.
		Structural changes often would lead to reduced costs in producing a product using inner source.
	<i>Knowledge issues</i>	The quality of the code increased as a result of the inner source implementation.
The use of domain experts from across the company can increase the product's quality significantly.		
<b>P r o c e s s</b>	<i>Communication</i>	CVS and associated repository tools, along with community communication tools (mailing lists, web forums, etc) were part of a common solution.
		Communication issues were common.
		Success factors that play a role in open source include simple communication mechanisms with a low learning curve.
	<i>Knowledge issues</i>	Computer mediated groups generate more ideas (Nunamaker <i>et al</i> , 1991).
		The breadth of ideas is a lot greater in asynchronous communications media such as is found in open source projects (Benbunan-Fich <i>et al</i> , 1999).
		Formalization inhibits innovation (Klincewicz, 2005).
		Open source development is co-evolution of code, knowledge and community.
	<i>Community</i>	There needs to exist a large degree of overlap between community values and governance structure.
		Motivation of company and contributors needs to be clear but crystal clear for the company.
		Open source can be described as constellations of communities who can work on what they choose rather than a formal set of tasks mandated by business objectives.
		As with implementing any change in a company and particularly open source changes to a company there is evidence to suggest that employees will resist change (Jaaksi, 2007).
	<i>Structure</i>	Forked code was a problem that effected support of the common infrastructure.
		Significant organizational structural changes were required.
		Structural changes often lead to reduced time to market for products using inner source.
		Boundaries are especially important, where developers can work and where contribution is not really requested.
		Open source can often be seen as an experimental approach to software development in that coordination takes place after action.

*Product* – We observed three aspects of a product that were questioned, domain area, knowledge issues and economic issues. Of these three the former two were for the most part considered unproblematic but economic issues such as the calculation of total cost of production and ownership is an area that needs further research as organizations are struggling with a way to come to an accurate estimate. Champions of Inner Source in organizations find it difficult to convince top management about the benefits of Inner Source

development if they cannot justify to at least some degree the cost factor involved.

Knowledge diffusion and innovation in the product are a true selling argument in favour of Inner Source use. The case studies indicate quite clearly that all the companies that chose to take the Inner Source route did so to enlarge the developer and knowledge base. Domain experts, through the inclusive nature of Inner Source, were able to contribute and guide project development which lead to better quality code, and applications that were closer to user requirements.

*Process* – Communication, knowledge, community and structure were observed to be important factors in Inner Source use. Again, we have some factors that are treated as if they are unproblematic like communication. This is how communication has been presented in the literature on Inner Source but we have

reason to believe that this aspect is closely related to governance and control issues and thus not as uncontroversial as appears.

The question of knowledge is raised again under process but the focus this time, as opposed to product knowledge where quality of code was the main concern, is on idea generation and linking this to idea dissemination and innovation. The process understanding of knowledge is much broader than the aspect looked under product. Here we show that code,

knowledge and community are related and indeed inextricably linked.

The issue of community is an important one in Inner Source and was mentioned repeatedly in the case studies examined, however, how do we define “community”? Should the “community” have certain traits in order to be able to support Inner Source? Hillery [19] found 94 different definitions of community with only one common element, that of people. Changes in what a community is today only serves to complicate any definition of community. The issue of community in Inner Source needs further examination and is an interesting area for future research. Questions such as how is Inner Source accepted in an organization, by whom, what is the governance structure of the organization and how does it adapt to Inner Source, and how are smaller sub-communities of developers assimilated if they are resistant to Inner Source are all pertinent and interesting avenues for future research.

From our initial examination of the area and this issue in particular we have observed the community to include all developers and staff with at least code interaction duties and possibly all employees. The issue of suitable communities of Inner Source did not arise as the case studies had an inclusive approach where everyone in the company had access, and an experimental approach where they envisaged a better product would be where everyone had input. The point was to implement the project and then evaluate the experience.

The most problematic, though interesting area under process that we found in our study is structure. This aspect is linked to all others under process but very closely to the community one. Each issue that we found related to structure has a positive and negative angle which implies how finely tuned the governance strategy of an Inner Source policy and company will need to be. Boundaries of the project and the organization are relevant and yet often blurred because it is difficult to enclose a community that feeds (and provides food for thought) on ideas. Employees leave an organization but take a ‘part’ of the knowledge with them, and new people join that then need to be *included*. Structure and organizational change will form a key part of our study in Phase II.

## 7. Conclusion

In conclusion we have examined a number of case studies involving various Inner Source solutions. We looked at the expected results of the Inner Source solution and the motivations behind the deployment. We then looked at the actual benefits and the problems

associated with the deployment. The issues of managing and organizing inner source were examined. These issues are especially important to Inner Source and as we have seen some critical changes were required in the successful implementation of Inner Source. In particular we have seen that to create innovation a flexible approach to managing is required and the organizational structure must undergo some key changes to foster growth of Inner Source. One particular finding is that organizational structure is the biggest issue that must be constantly examined when implementing an Inner Source solution.

As can be expected this initial exploration of the nature of Inner Source has brought forward more research questions such as the exact extent and metric of our initial findings, particularly in the area of reported faster development cycles and software reuse. Our paper also clarified what we understand to be Inner Source principles. We looked at emerging principles in terms of how they relate to open source and also how they relate to traditional software development. These principles help to characterize the Inner Source phenomenon more clearly; set out the scope of our future research; and also help us to explore the potential cross over of Inner Source principles beyond software. The change occurring in the corporate world in order to leverage the successes of open source is a phenomenon not limited to software and this is another aspect we and other researchers are currently exploring.

Our study is of course naturally limited by the nature of the topic have a limited number of implementations to study, and more so by the small sample size that can be attributed to the design of the exploratory pilot study. However, we feel that we have achieved what we set out to do by characterising Inner Source solutions in existence today.

## 10. References

- [1] J. Herbsleb "Global Software Engineering: The Future of Socio-Technical Coordination". in *Future of Software Engineering FOSE '2007*, IEEE Computer Society, 2007 pp. 188-198.
- [2] E. Raymond “*The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*”, O’Reilly & Associates, Sebastopol, California, 1999.
- [3] J. Dinkelacker, P. Garg, R. Miller, and D. Nelson, “Progressive Open Source.” In *Proceedings of the 2002 ACM International Conference on Software Engineering (ICSE’02)*, ACM, 2002 pp. 177-184,.
- [4] V.K. Gurbani, A. Garvert, and J.D. Herbsleb, “A Case Study of Open Source Tools and Practices in a Commercial

Setting.” *In Proceedings of the 5th Workshop on Open Source Software Engineering*, ACM, 2005 p. 24-29.

[5] K. Crowston, B. Scozzi, “Open source software projects as virtual organizations: Competency rallying for software development.” *IEE Proceedings Software*, 2002, 149(1), 3-17.

[6] M.J. Gallivan, "Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies," *Information Systems Journal* (11:4) 2001, pp 277-304.

[7] J. Dinkelacker, P. Garg, “Corporate Source: Applying Open Source Concepts to a Corporate Environment. “HP Technical Report 2001.” [http://www.hpl.hp.com/techreports/2001/HPL-2001135.html?jumpid=reg\\_R1002\\_USEN](http://www.hpl.hp.com/techreports/2001/HPL-2001135.html?jumpid=reg_R1002_USEN)

[8] G. Walsham, “Interpreting Information Systems in Organizations”, *Wiley*, Chichester, 1993.

[9] G. Walsham, "Interpretive case studies in IS research: nature and method," *European Journal of Information Systems* (4), 1995, pp. 74-81.

[10] A.S. Lee, "Electronic Mail as a Medium for Rich Communication: An Empirical Investigation Using Hermeneutic Interpretation," *MIS Quarterly* (18:2), June 1994, pp. 143-157.

[11] R. K. Yin, “Case Study Research, Design and Methods”, 3rd ed. Newbury Park, *Sage Publications*, 2002.

[12] I. Benbasat, D.K. Goldstein, and M. Mead, “The Case Research Strategy in Studies of Information Systems”, *MIS Quarterly*, 1987 11:3, pp. 369-385.

[13] T.V. Bonoma, (1985) “Case Research in Marketing: Opportunities, Problems and a Process” *Journal of Marketing Research* 1985 (22), pp. 199-208

[14] C. Melian, “Progressive Open Source”, PhD Thesis, *Stockholm School of Economics* 2007.

[15] M.J Garcia, and E.W Steinmueller (2003) “The Open Source Way of Working: a New Paradigm for the division of labour in Software development?” INK Open source working paper No.92, SPRU Science and Technology Policy Research, University of Sussex, available at: <http://www.sussex.ac.uk/spru/1-6-1-2-1.html>, last accessed 15/03/09.

[16] A. Capiluppi, and M. Michlmayr, “From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects” in IFIP International Federation for Information Processing, Volume 234, Open Source Development. Adoption and Innovation, eds. J. Feller, Fitzgerald. B., Scacchi, W., Sillitti, A. Boston: *Springer*, 2007 pp. 31-44.

[17] J. West, and C.S. O’Mahony, ‘Contrasting Community Building in Sponsored and Community Founded Open Source Projects’, 2004 available at: <http://opensource.mit.edu/papers/westmahony.pdf> , last accessed 06/03/09.

[18] B. Fitzgerald, “The transformation of open source software”. *MIS Quarterly* 30, 3 2006, 587--598.

[19] G.A. Hillery, G. A. (1955) "Definitions of Community: Areas of Agreement", *Rural Sociology*, 20 1955 pp. 111-123.