

Open Source Software Ideology, Self Interest and the Crowd

Tinesh Babu

*MSc Analysis, Design and Management of Information Systems
Information Systems and Innovation Group
Department of Management
London School of Economics and Political Science*

KEYWORDS

Open Source Software
Free Computer Software
Technology Innovation
Ideology
Economics of Technology
Professional Open Source
Software

ABSTRACT

Researchers have tried to analyse and represent the organisation and development within the Open Source Software (OSS) realm using various socio-technical models. In this literature review, I have tried to compare and contrast these various models. Also, this literature review looks into various reasons that have been proposed by researchers for the motivation of individual programmers to contribute towards OSS. This paper attempts to compare how both ideology and self-interest act as a motivating factor for contribution. With private companies participating and investing heavily in Open Source projects, I try to identify their motivation and benefits from participating in OSS projects by looking into past studies in this area.

Introduction

The Open Source Software (OSS) arena has been undergoing interesting changes for some years now. Found on the basis of the Free Software Movement and knowledge sharing, developers or volunteers contribute willingly and wholeheartedly for no apparent monetary gain. Recently, the arena has seen the influx of corporate giants based on pure capitalist ideas, focused mainly on profits, contributing considerably to OSS. The main reason why Open Source Software has been able to grow to its current levels and challenge Closed Source Software in open market is attributed to its high quality, which is comparable or at times better than closed source software. OSS communities have been able to attract highly skilled programmers to contribute voluntarily in large numbers towards the development of software. Reasons attributed to this include ideology, a quest for reputation, career growth etc. (Steward and Gosain 2006; Markus et al. 2000). Though not explicitly evident, economic benefits too have been mentioned as another reason.

At the same time, more and more contributions to the OSS realm are coming from paid software professionals as part of their work within their organisations. In fact, a majority of the contributions towards embedded Linux kernel nowadays comes from paid programmers as part of their work. The libertarian values which formed the basis for OSS

could be eroded by the entry of private firms into the OSS realm, but on the other hand, the irreversible nature of the open source code adds to the knowledge base of the contributors, both independent and paid.

The organisational structure of OSS communities and their development models have been of great interest for researchers for some time now. The complex social phenomena and interaction within the OSS communities have necessitated a socio-technical approach in analysing the OSS development and organisation. While the metaphoric cathedral and bazaar models had been proposed by Raymond (1999) to analyse the OSS development model, Actor Network Theory (ANT) has been proposed to better capture the complex social characteristics within the OSS community. The complex social structure and communication within the OSS communities provides a compelling case for analysing OSS based on the Social Shaping of Technology (SST), ANT and the Social Construction of Technology (SCOT).

Motivation: Ideology and Self Interest

Open Source Software production contrasts greatly in contrast to closed source software production, especially in terms of participation and contribution of developers. While the closed source software producers pay a salary to their developers, the OSS participants contribute without any financial gain, and yet produce software of high quality. This unique aspect of OSS has prompted many to ask the question as to why developers contribute, with many

Corresponding Author
Email Address: tinoos@gmail.com (T. Babu)

researchers reaching conclusions such as the altruism, reputation and ideology of the OSS development community (Markus et al. 2000; Steward and Gosain 2006). Ideology and tenets of OSS ideology are shown to have a positive impact on the trust and communication within the OSS project (Steward and Gosain 2006). Trust and strong communication are extremely important to OSS, in order to bring about better coordination among programmers within the community, but also to keep defections to a minimum as well as to attract new programmers to the project.

Yet, OSS ideology and altruism does not give a complete picture regarding the motivation of individual programmers. Studies have shown that most developers in OSS projects are not driven by ideology or altruism but rather by self-interests (Hertel 2007). Apart from aspects like gaining reputation amongst peers, wanting to learn new skills or creating better software for one's own use, economic benefits have been mentioned as a possible motivator for an OSS contributor. But, seeing economic benefits as a motivating factor for individual programmers is ambiguous as these volunteers contribute to the project without any immediate visible financial gain. It has been suggested that economically, compared to short term benefits or immediate gain, programmers have good long term incentives.

By being part of a successful open source project, the programmers are able to show outsiders their exact contribution, the complexity of the problem and the approach taken towards its solution. The ability of the programmer to perform and follow through also becomes evident to outsiders, as the programmer works on his own on a sub project of the OSS project (Lerner and Tirole 2005). These acquired qualities and skills make the OSS programmer more valuable to future employers. The results of an econometric study on contributors to the Apache project, which draws upon a wide range of project records, has been consistent with the view that individual programmers/volunteers to OSS projects do benefit directly (Lerner and Tirole 2005; Hann et al. 2004). Finally, long term benefits for programmers are stronger under three conditions: "1) the more visible the performance to the relevant audience (peers, labour market and venture capital community); 2) the higher the impact of effort on performance; and 3) the informative the performance about talent" (Lerner and Tirole 2005: 103). Hence, it is quite evident that the economic benefit, even though delayed, plays a major factor in attracting individual programmers towards OSS projects. Whether the motivating factor is ideology or self-interest, the fact that the programmer is also a user means that there should not be any impact on the quality of the software produced.

The Crowd: Social Organisation within OSS

The high quality of OSS is attributed to its ability to attract large numbers of volunteers to participate and improve the product. To put it in Raymond's words, "[p]erhaps in the end the open-source culture will triumph not because cooperation is morally right or software "hoarding" is morally wrong but simply because the closed-source world cannot win an evolutionary arms race with open-source communities that can put orders of magnitude more skilled time into a problem" (Raymond 1999: 54).

This aspect could be compared to a socio-technical approach towards the OSS communities, in particular, the Social Shaping of Technology (SST) for the development of open source software. From an engineering point of view, Raymond (1999) suggests that OSS developers tend to follow a Bazaar model in contrast to the Cathedral model followed by closed source software developers and early OSS developers. The focus here is to release early and often and to delegate as much as you can (Raymond 1999). But it also lays a great deal of emphasis on the project and its leaders in order to attract a large number of contributors and beta testers. It is the ability of OSS to bring together a large team of dedicated developers that actually contribute to the high code quality of the software. In what Raymond (1999:30) calls the Linus' law, "[g]iven a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone. Or, less formally, 'Given enough eyeballs, all bugs are shallow'."

In return, it is this high quality and reliability that fundamentally contribute towards the popularity and growth of Open Source Software. It has to be noted here that since the developers are also users of the program, the motivation for contributing towards development and error correction remains stronger in comparison to a paid software professional working on closed source software. This could be evaluated as a socio-technical model of organisation that fits perfectly with the Social Construction of Technology (SCOT), wherein the view or requirement of the user needs to be taken into consideration by the developer, which is satisfied within OSS with the developer himself being a user.

It is the ability of OSS to attract large numbers of volunteers, either as developers or testers, that has enabled it to produce high quality software, comparable or even better than closed source software. It's been puzzling for researchers, how OSS developed without any use of conventional software development methodologies and, being more like an amorphous, bazaar like and structure less organ-

isation (Dahlander, 2007), can produce products of such high quality, even if it can attract a large team size. It has been suggested that it is the trust (both cognitive and affective) in the team leader, output task or project completion as well as communication quality, which have their basis on OSS norms, beliefs and values, that has contributed to large team sizes and committed efforts for individual projects within OSS (Steward and Gosain 2006). This in itself is even more puzzling considering the fact that a project leader or team leader within the OSS realm does not have any actual control over participating developers. OSS communities establish mechanisms for reporting contributions from members to be evaluated, monitored and integrated into the community. They have tools like discussion boards, forums etc. that help volunteers to participate in discussions and to develop formalized structures in which certain individuals within the community have more formal authority than the rest (Dahlander 2007). The social organisation within the OSS project/community determines the structure of formal authority and leadership progression, which can range from a completely democratic process to that of a benevolent dictator.

It has been suggested that OSS communities work in a meritocratic fashion, a system of government based on rule by merit rather than by wealth, class, political power or the social position of the individual (Dahlander 2007; Kogut and Metiu 2001). The Apache web Server project is an example of a meritocratic democratic organisational setup, wherein to be a leader one has to be a major contributor to the project. The group of leaders, called a project management committee, decides by vote (Apache 2010). For instance, code changes to the web server can be requested by any developer via a mailing list, which is then evaluated and put to a vote.

Not all OSS communities/projects work in a meritocratic fashion. Despite contributing about 2% of code to the Linux kernel, Linus Torvalds remains its ultimate authority (Dahlander 2007). He, along with his trusted lieutenants, controls any changes or modifications that need to be done to the Linux kernel code and, hence, is considered a benevolent dictator. While an individual contribution plays a major role with respect to becoming a leader, it is also the ability to form connections or ties within the community - especially with the current leadership - that plays an important role in rising through the ranks of the community (Dahlander 2007).

Open Source Software development and its organisation could be looked through another socio-technical model, namely Actor Network Theory (ANT), a view that in itself contrasts with the traditionally perceived bazaar model. The version control soft-

ware used in the OSS development could act as an agent in the actor network of OSS development (Cornford et al. 2005), while strong mentorship and encouragement can act as a factor towards attracting new developers (Steward and Gosain 2006), in called allies in ANT. ANT considers the social and technical as inseparable and analyses both people and technology involved with the same conceptual apparatus (Howcroft et al. 2004). The OSS community and the network between the developers within the community, the tools they use, the controversies and dissents about OSS, which sometimes even lead to the forking of the project, all of which together form a complex social phenomenon, can be analysed using ANT. The powerful and elaborate vocabulary offered by ANT for analysing the development of a project, through its concepts of enrolment, mobilization, dissidence and inscription, helps to draw greater attention to the process of translation and the political implications with respect to OSS development and organisation (Ramiller 2007; Cornford et al. 2005).

Open Source and the Business World

Commercial software developers, including ones who were previously strong proponents of Closed Source Software, too have entered the OSS arena. Primarily with a view of increased revenue and profits, their strategies and approaches vary greatly in comparison to developer communities and individual programmers. Some of these companies have seen an opportunity for revenues through training, consultancy or support of OSS packages (Watson et al. 2005; Fitzgerald 2006), while others follow the loss-leader strategy, where the actual package or product is in itself distributed for free with the goal to enlarge the market for allied closed source products and services (Fitzgerald 2006). Companies in OSS also have an additional advantage when it comes to recruiting. They are able to tap into an already existing pool of developers from the OSS communities lowering training costs and quality risks (Watson et al. 2005).

Additionally, companies have adopted OSS products, due to their high reliability and lower costs (often zero cost), as a basis upon which they have created their own platforms to deliver value added services within their business domains, which also ensures that the Open Source remains invisible to their customers (Fitzgerald 2006). Apart from adopting the above strategy, companies like Google, Amazon, Sourceforge.com etc. also customize open source products to suit their internal needs. By the same token, they do not face any problems regarding noncompliance of OSS licences, since there is no redistribution of this customized Open Source Software (Fitzgerald 2006). Open Source firms also ben-

efit from lower Research and Development (R&D) costs due to free riding (Llanes and Elejalde 2009) driving down the total expenditure for OS firms bringing a positive impact on profits.

Conclusion

Open Source Software has been able to attract large numbers of developers to its projects reaching levels that cannot be matched by any closed source developer (Raymond 1999). This is due to a variety of reasons ranging from an adherence to an ideology, quests for reputation, career growth (Steward and Gosain 2006; Markus et al. 2000) to a work ethic to achieve the common goal of producing high quality software. Analysing OSS through different socio-technical models has helped to understand how the OSS communities organise themselves in order to form common goals, clear leadership roles and control mechanisms even in the absence of formal organisational structures. While an analysis of OSS development using ANT is certainly helpful, it needs to be further probed and researched.

On the one hand, one can argue that the ideology inspired, altruism-based development of OSS has given way to self-interest-based development and participation, which has not only sustained OSS, but also helped prosper both in terms of attracting more participants/users and applications. On the other hand, it can also be argued that since all new additions remain open, it helps the social commitment as originally envisaged in F/OSS, even if it is not the primary intention of the developer. The same holds true for open sourcing of code by private firms and increased contributions by paid programmers to OSS projects as part of their work. Any contribution to OSS, whether by private firms or individual contributors, will always add to the knowledge base of OSS.

What effects the entry of private firms has on OSS structure and development will need to be further researched. The effect of increased contributions by paid programmers to OSS also needs to be further researched, especially in the context of the effects it can have on bug/error reporting within OSS. Linus' law of a sufficiently large number of eye balls being able to find all errors is founded on the assumption that the developers and testers are first users of the same software. In most of the cases, the paid programmers themselves are not users of the OSS project they are contributing to, hence their motivation

to improve the software or their understanding of the software is in itself in question and needs to be further investigated.

References

- Apache (2010). The Apache HTTP Server Project. http://httpd.apache.org/ABOUT_APACHE.html (accessed 13 December 2010).
- Cornford, T., Ciborra, C. and Shaikh, M. (2005). Do penguins eat scallops? *European Journal of Information Systems*. 14(5) pp.518-521.
- Dahlander, L. (1993). In the club: Human and social capital of leaders in Free and Open Source Software. *Academy of Management*. 9(8) pp.1-6.
- Fitzgerald, B. (2006). The transformation of Open Source Software. *MIS Quarterly*. 30(3) pp.587-598.
- Hann, I.-H., Roberts, J., Slaughter, S. and Fielding, R. (2004). An Empirical Analysis of Economic Returns to Open Source Participation. Working Paper, Carnegie Mellon University.
- Hertel, G. (2007). Motivating job design as a factor in open source governance. *Journal of Management & Governance*. 11(2) pp.129-137.
- Howcroft, D., Mitev, N. and Wilson, M. (2004). What we may learn from the social shaping of technology approach. In Willcocks, L. and Mingers, J. (Eds.) *Social Theory and Philosophy for Information Systems* (pp.329-371). Chichester: John Wiley.
- Kogut, B. and Metiu, A. (2001). Open-Source software development and distributed innovation. *Oxford Review of Economic Policy*. 17(2) pp.248-264.
- Lerner, J. and Tirole, J. (2005). The Economics of Technology Sharing: Open Source and Beyond. *Journal of Economic Perspectives*. 19(2) pp.99-120.
- Llanes, G. and Elejalde, R. D. (2009). Industry Equilibrium with Open Source and Proprietary Firms. Harvard Business School Entrepreneurial Management Working Paper Series, Working Paper No. 09-149.
- Markus, M. L., Manville, B. and Agres, C. E. (2000). What Makes a Virtual Organization Work? *Sloan Management Review*. 42(1) pp.13-26.
- Ramiller, N. C. (2007). Constructing safety: System designs, systems effects, and the play of heterogeneous interests in a behavioural health care setting. *International Journal of Medical Informatics*. 76 pp.196-204.
- Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*. 12 pp.23-49.
- Steward, K. J. and Gosain, S. (2006). The Impact of Ideology on Effectiveness in Open Source Software Development Teams. *MIS Quarterly*. 30(2) pp.291-314.
- Watson, R. T., Wynn, D. and Boudreau, M.-C. (2005). JBOSS: The Evolution of Professional Open Source Software. *MIS Quarterly Executive*. 4(3) pp.329-341.