



AN ANALYSIS OF INTELLECTUAL PROPERTY DEVELOPING TOWARDS OPEN SOURCE SOFTWARE

Chavan Ashwini M.

Research Scholar Kalinga University

Dr. Rupak Sharma

Asst. Prof

Declaration of Author

I hereby declare that the content of this research paper has been truly made by me including the title of the research paper/research article, and no serial sequence of any sentence has been copied through internet or any other source except references or some unavoidable essential or technical terms. In case of finding any patent or copy right content of any source or other author in my paper/article, I shall always be responsible for further clarification or any legal issues. For sole right content of different author or different source, which was unintentionally or intentionally used in this research paper shall immediately be removed from this journal and I shall be accountable for any further legal issues, and there will be no responsibility of Journal in any matter. If anyone has some issue related to the content of this research paper's copied or plagiarism content he/she may contact on my above mentioned email ID.

ABSTRACT

The use and adoption of Open Source Software (OSS) initiatives in the wild are increasing. Not only are many software development firms joining the phenomenon, so are governments. In particular, developing countries still face a multitude of problems in use and acquisition of software, to which OSS could be a potential solution. To them, delivery of service would be more important than profit, which would otherwise be hampered by the cost of acquiring proprietary software. To explore this topic, we used literary review to understand the software landscape of developing countries, as well as find out how OSS could be used to improve their standing in the information and communication technology (ICT) scene. To this effect, this paper researches the topic from three aspects, language, cost and adoption of software in these countries, and how OSS could be used as an alternative to tackle these topics. Results of this study point out that OSS can not only improve the availability of software in the local language of developing countries, but it can also provide cost effective solutions. However, to reach better OSS adoption rate governmental involvement through implementation of policies is required. This new community based software development model, instead of relying on the conventional proprietary model of limited access, invites programmers globally, to freely copy, share, and modify the software. It is a misconception to believe that the general approach of open source software (OSS) towards IP laws and focuses on specific issues that emanate from the interaction of the OSS model with the existing intellectual property rights structure. The open source movement necessitates scrutiny; more than just being a new-fangled approach, it catalysis debate regarding both the mode of software production and its protection. It is being used to propel arguments to revisit intellectual property jurisprudence. After all, the intention of the intellectual property-software system is to catalyze innovation and ultimately serve the society.

KEYWORDS: *Open source software, intellectual property rights, license, proprietary.*

INTRODUCTION

In today's software development scene, there are increasing numbers of Open Source Software (OSS) projects and initiatives in the wild. And, without a doubt, these numbers will only increase. While many software firms are joining the phenomenon to lower development costs and gather OSS community feedback, all the while profiting indirectly from the sales of related products or services, movements like these will, regardless of intentions, make software more available to those that need them. Governments too are exploring OSS options in order to increase flexibility, reduce costs and improve reliability. This is particularly important today as countries need to stay on top of their game in order to keep up with the advancements made in Information and Communication Technology (ICT), which in their paper argue is a struggle in developing countries. They argue that proprietary software is often times just too expensive for these countries to afford, as well as English proving to be too big of a language barrier for the major part of their population. This means that developing countries need to play a catch-up game just to remain competitive in today's world.

The institution of open source has its roots in an ethical rebellion of sorts by software programmers expressed in the form of sharing source code of computer programs. The cause was subsequently taken up by academics and practitioners alike, and eventually this model snow balled into a

parallel regime in the software development landscape.

Open source software represents a paradigm shift in the field of software development. As opposed to the generically referred 'conventional,' 'proprietary,' 'closed,' source code model, the open source software model emphasizes on unrestricted accessibility to the source code of the program. Also, unlike, proprietary software, open source code development is not a solitary or a closed group task – it is a community based development model adopting the 'bazaar style,' as opposed to the 'cathedral' approach. The open source development model is not subject to prejudices based on maturity, education or experience. Also, there is no need to assemble the community at one physical place, virtual interaction is sufficient. Thus, the open source model allows larger groups to interact, and increases accessibility to many more resources, while keeping the transaction costs at a minimal.

GOVERNANCE STRUCTURE OF OPEN SOURCE SOFTWARE

Traditionally, copyright law was available to protect literal and the patent law, mechanical. Being a written work with a utilitarian character, computer programs pose difficulty in categorization in the present library of IP protection. Despite the paradox, the current IP regime, arrived at an accommodation, and protects various components of computer software

separately. Trade secret law was the traditional vehicle of software protection; it can protect secrets embodied in or implemented through software. While, copyright chosen as the legislative vehicle protects the literal expression of software; patent protection for software has grown doctrinally and essentially protects the technological expression of software. Trademark protection, moral rights and design protection laws along with technological protection measures and licensing too are applied. It is not necessary that all the forms of protection would be exercised with every software; developers may choose to rely on none, some, or all of them. Interaction of the traditional intellectual property structure and computer software generated the proprietary model of software development, which was the veritable apotheosis for over two decades. Laments increased that traditional IP protection in software were steadily whittling down the public domain. The key critique of the traditional manner of application of IPR to software is that it is better suited for protecting, than for facilitating relinquishment of rights. Open source philosophy when applied to software re-imagines the interaction and interpretation of various IP laws.

OSS AS AN ALTERNATIVE:

1. Open Source driven localization: Due to the high cost of proprietary software to developing countries, localization efforts may not be cost-effective to pursue in commercial software. However, OSS and FOSS ideologies provide a platform on

which software development communities of these locales may start their own efforts. McNaughton claims that most researchers agree that OSS is preferred for localization due to its open nature. For this to happen, only a handful of individuals with the right capabilities are needed to provide a localized version of any OSS. There have already been a number of OSS localization efforts. For example, in Urdu Linux Localization Project (ULLP) was created using FOSS principles in order to make it available in the local language. According to them, not only was Linux chosen due to its licensing terms (GNU-GPL), it was also chosen because the operating system has gone through the internationalization process. In fact, in Mongolia students having access to software in their own language has helped boost their interest in computers, and thus development of software). All the while, in India, a FOSS project exists that aims to provide optical character recognition (OCR) for Indian languages. OSS adoption like this can be seen even on governmental level, as such is the case in India, the government has made” wide efforts to make software available to millions of users through localized versions of OSS”. This is compounded upon by the fact that localization of FOSS can also promote local talent through participation. By having more common FOSS-products available in local languages their usability can be increased, which in the case of Tanzania would allow people without knowledge of English and Swahili to also try them. This would further improve the country's IT development scene, in addition

to making software more available in local languages.

2. Reduced cost: Open source software saves costs for public organizations: licensing, vendor lock-in and high switching cost can all be reduced by using OSS. Because of this, OSS is usually treated as free of charge software and not as “free as a bird” because of free distribution. In this respect, the use of OSS would reduce costs, because it does not require any third party intervention to use. Open source software is a good solution for a developing country due to its low cost of acquisition. To establish enterprise software infrastructure OSS can be seen as a regular technology and this will encourage economic expansion in developing countries. However, software implementation costs can be quite high. They argue that in OSS money is spent more on services, migration and education of users while in proprietary most of the money goes into licenses, keeping the total cost of ownership of software about the same. However, they continue to say that with OSS the money remains inside the country, as opposed to going to the software vendors. Yildirim & Ansal (2011) voice similar concerns, noting that OSS is rather a “new system deployment” than a replacement for proprietary software. Additionally, when evaluating the long-term economic effect of switching costs, the flexibility of user freedom with the software should also be considered.

3. Adoption of OSS: Benefits of adopting OSS over commercial closed-source software is initial low of cost acquisition,

due to ability to run on older hardware, lower need to upgrade the hardware and support from OSS community over paid services. In their article that the cost shifts from initial acquisition price into modifying the product in to local needs and that foreign company using similar business model can be useful asset in helping to execute these changes. OSS has benefits on not being dependent on foreign software suppliers. One concern relating to OSS adoption that security may be lacking compared closed-source software that gets periodically updated. Especially if the OSS software is modified. Also, according to them government agencies have acknowledged the issue. However their ways to attempt fix it has not been effective in term of cost and performance. One solution for easing the adoption of OSS software is NSI, in which being conjoined with lead from government and by policies made by it for instance using OSS within the government itself, can lead to larger scale and success of adoption OSS software. In his article that flexibility when it comes to government IT policy is necessary and as example he uses India whose IT acquisition policy requires flexibility that their commitment to OSS can provide. Governments in ASEAN have made macro level decisions, by making policies mandating usage of OSS software. Which in their eyes gives them more power in the IT sector and reducing dependency on foreign software providers at the same time? Also those in these countries despite government’s commitment OSS software commercial software such as ones from Microsoft are used, while ignoring the costs,

mainly due to piracy. They also mention that companies providing commercial software are not going to give up their market share in these countries to OSS without of fight.

SOFTWARE: PROTECTION AND MANAGEMENT

Debate is raging as regards intellectual property and software. Suggestions vary from application to non-application of property rights to software; from correct interpretation, to amendment to outright replacement of property laws with alternative models of *suigeneris* software protection. Fuelling this debate is the current success of open source software, arguably without the conventional mode of software protection.

As stated earlier, the open source faction staunchly opposes the current software patents regime. Essentially most open source concerns emanate from ineffective use and regulation of the patent system coupled with legal issues in context of software patents. A major bone of contention is the patent term which though considerably shorter than the copyright term is much more enveloping in effect due to its monopoly nature. However, though the potential for harm is great, it does not mean that ideas contained in software patents should not be protected. Patents in software are a relatively new development on the intellectual property front. As with other developments, the rough edges of this development too need to be smoothed out. Rather than completely eliminating patents as a medium of protection for software, efforts should be

focused on a better system with higher standards of examination. Instead of complete radical substitution, mere supplementation and rectification of the patent system addressing these concerns is a possibility.

When open source supporters protest against software patents, an inevitable question that arises is whether copyright protection alone is adequate to protect computer software? Though the copyright regime has its fair share of supporters, critics, think-tanks and government bodies are largely of the view that copyright law by itself is an inadequate mode of protection for software and could actually be much worse for the industry and open source. US Judiciary too noted this deficiency. Primarily, copyright law by its very nature protects expressions and not the underlying ideas. In case of computer software, it transitions to protection of the expression but not the underlying functionality. Hence, the essence of computer software is open to analogous development. This in turn allows several expressions of the same innovation to exist. Some critics view this conflict as an opportunity to revisit copyright jurisprudence and redraw the lines to properly demarcate authorship issues, and resolve the idea-expression dichotomy, at least in context of software programs.

While some critics cite the need to revisit intellectual property jurisprudence, others challenge the very existence of property rights in software development. The open source movement is a viable mode of software development, and their argument

that property rights should not apply to software may hold merit. However, the real question is whether absence of property rights would be incentive enough for software development. Moreover, a fundamental question to be answered is whether the open source movement would be an adequate replacement? The open source community may not be ideally suited to provide the requisite innovation impetus. Although many contributors appear to participate in open source software development for reputational rewards, it is unclear whether such intangible benefits are a sufficiently effective motivation to drive large-scale efforts. More than initiation, sustaining of such projects is a big question. The debate has its share of supporters and detractors. Moreover, open source penetration is not sufficient and an overnight radical change could well mean disaster. Additionally, the software industry is not an island; any change in the protection regime, would have a cascading effect on several other industries. Furthermore, what needs to be considered is that if there is a complete replacement of the proprietary structure with the open source model, then the corresponding business model would alter the nature of the software industry, from that of software creation to software servicing.

It cannot be denied that a complementing stance needs to be acquired to accommodate the OSS model. The open source regime has acquired a stature which cannot be just brushed aside as an anomaly. Complete removal is not a good option; the open source model provides a good counterbalance in the software industry to

the proprietary model. Besides preventing market dominance and monopoly it also allows for greater variety; companies are able to combine the best of both worlds. Also, regularizing the open source model allows its proper regulation. At the end of the day, it is not desirable to establish an exclusive system where only one of the policy models exists. Adjustments would be needed to harmoniously integrate the open source model with the proprietary model in the intellectual property regime. Any regime not doing so 'runs the serious risk of remaining a vestige of twentieth-century IP regimes focused on rights of exclusion rather than enabling and encouraging rapid software development.'

Instead of focusing on the debate as a means, the focus should be on the ends. The need of the day is efficient and cost effective software. It is irrational, not to encourage any capable production mode, whether it be proprietary or open source. Both have their positive attributes and their shortcomings. Moreover, it is difficult to assess whether either model would be more successful without the influence of the other. What needs to be realized is that as things currently stand, both the modes cannot be done without. 'Essentially, both the approaches require balancing of commerce and user independence. To completely dismiss one approach in favour of the other is to abandon a mature method of fostering innovation for another that is still immature.' Furthermore, courts and market inertia will force both the open source and proprietary models to coexist. 'Given the uncertainties in determining which kinds of

endeavors can be safely left to open innovation, it is likely that a dual system will be operative in many technological fields.' Hence, a harmonious construction of software protection is the best option.

CONCLUSION:

Software programming tends to be based on de facto standards and the case of de facto standards is that they find favour only up to the time they are the best available solution. Earlier computer programming was done on one-to-one basis and cyberspace was yet to be established as a medium of connectivity. However, that is not the case anymore. Software is rarely individually tailored, but mass-produced, and the existence of the Internet makes global collaboration extremely easy. Furthermore, programming itself has become relatively easier. Not taking advantage of such a scenario would amount to restricting infrastructural growth. Open source development has become a practical alternative to more traditional proprietary production schemes for software. The open source faction is using the open source method to renegotiate and properly channelize electronic information governance. Community involvement has been the leading light as regards formation and management of the open source philosophy. It turned into an extra-judicial arm and filled a perceived gap in software management which was not being addressed thus leading to the founding of OSS. However, despite the growing use and acceptance of OSS, the legal rights afforded to open source developers remain largely

unknown. Furthermore, considering the dynamism of the software field, the involved community too needs to change its outlook and evolve with the times. The emergence of splinter factions within the open source philosophy bolsters this argument. It is true that the open source groups lobby against patent protection for software, but they need to take a more proactive role and lobby for overhaul of the software protection regime altogether.

REFERENCES:

- [1] Jokonya, O. (2015, January). Investigating Open Source Software Benefits in Public Sector. In System Sciences (HICSS), 2015 48th Hawaii International Conference on (pp. 2242-2251). IEEE.
- [2] Phongpaibul, M., & Aroonvatanaporn, P. (2015, November). Standardized cost estimation in Thai government's software development projects. In 2015 International Computer Science and Engineering Conference (ICSEC) (pp. 1-6). IEEE.
- [3] Sooryanarayan, D. G., Gupta, D., & Rekha, V. S. (2014, December). Trends in Open Source Software Adoption in Indian Educational Institutions. In Technology for Education (T4E), 2014 IEEE Sixth International Conference on (pp. 249-252). IEEE.
- [4] Sukhoo, A., Soobron, M., Soodin, R., Hawabhay, R., & Beerbul, S. (2013, May). Open source software adoption in Mauritius. In IST-Africa

- Conference and Exhibition (IST-Africa), 2013 (pp. 1-10). IEEE.
- [5] Tomar, V., & Bhatia, M. (2014, December). Software Localization kit for Indian mass. In 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) (pp. 000247-000255). IEEE.
- [6] Azzi R, CPR: How Jacobsen v Katzer resuscitated the open source movement, *UniversityofIllinoisLawReview*, 2010 (4) (2010) 1271.
- [7] Zhussupova, A., & Rahman, A. A. (2011, September). Open source software adoption in public organizations of Kazakhstan. In Open Systems (ICOS), 2011 IEEE Conference on (pp. 417-422). IEEE.
- [8] Yildirim, N., & Ansal, H. (2011). Foresighting FLOSS (free/libre/open source software) from a developing country perspective: The case of Turkey. *Technovation*, 31(12), 666-678.
- [9] Pranic, D., & Požgaj, Ž. (2010, May). Usage of open source software in public administration of republic of croatia. In MIPRO, 2010 Proceedings of the 33rd International Convention (pp. 1316-1321). IEEE.