Competition against Open Source community*
Very preliminary version

Edmond Baranes† Cuong Hung Vuong‡ Mourad Zeroukhi§

April 21, 2019

Abstract

This paper provides a simple theoretical framework to analyze competition between a commercial firm and a community of voluntary developers of an open source software (OSS) in the same market. The firm offers proprietary software at a positive price level while the OSS is provided free of charge. The firm can also interact with the OS community in the labor market by strategically setting the wage level of developers who work for the firm. A developer however could join the OSS community without monetary reward due to the signalling incentive. Therefore, we employ the standard Hotelling's style to account for the heterogeneity among the users as well as the developers. Additionally, the OS community suffers from forking that generally causes a weakening of the OSS project. In the baseline model, we can show the optimal wage might be increasing in the forking level. We then extend our analysis to examine the competition in three dimensions. First, we evaluate the effectiveness of a public policy that is imposed to increase coordination of the OS community. Second, the OSS is sold by a service provider that brings higher utility for users. Third, the firm could invest and contribute to the OS community to enhance the software differentiation.

JEL Codes: L13, L51, L96

Key words: Open Source Software, Differentiation, Competition, Forking

1 Introduction

Open Source Software (OSS) has become a serious competitor for the dominant proprietary software (PS). Some successful OSS such as Linux, Apache and Sendmail are

*We have benefited from helpful discussions with Eric Malin, Dominique Torre, Jean-Christophe Poudou and Jean-Benoît Zimmermann.

†MRE & University of Montpellier; edmond.baranes@umontpellier.fr.

‡MRE & University of Montpellier; hung-cuong.vuong@umontpellier.fr.

§CREM, University of Rennes 1

challengers (leaders even in the case of Apache) in their respective market segments. For example, the market share of the operating system Linux is around 21.2% behind the traditional leader in the sector, Microsoft Windows with around 73.9% (IDC, 2010). The Apache web server dominates its market with 54.56%, leaving Microsoft IIS (Internet Information Server) behind with 24.27% (Netcraft, 2010). These successes of OSS have lead to an increasing interest of the proprietary firms to know about the strategic behaviors to adopt for competing against OS Community. This interest is best summarized up in the following statement from an internal email written by Steve Ballmer, the Microsoft CEO in 2003 "*Non-commercial software products in general, and Linux in particular, present a competitive challenge for us and for our entire industry, and they require our concentrated focus and attention*".

In this paper, we examine how a proprietary firm could compete against an OS community both in the labor market of programmers and in the product markets under preference heterogeneity and in the presence of forking. Precisely, in the baseline model, there is a mixed duopoly model in which a proprietary firm competes against an OS community in a two stages game. At the first stage, the firm and the OS community develop the quality of their software products by hiring programmers on perfectly competitive labor market. The firm pays the same wage to the programmers while the OS community offers them the possibility to signal their abilities, which leads to non-pecuniary rewards (e.g., ego gratification, peer recognition, personnel learning and enjoyment from programming). The quality of the software product depends on the number of programmers working on. At the second stage, the users evaluate the two products and decide whether to buy the PS or to acquire the OSS for free. The users and the programmers are heterogeneous with respect to their preferences for the firm and OS community on their respective markets. Moreover, we assume that the OS community suffers from forking in software developing, which decreases the OSS's quality.

We can show that the optimal wage depends on the levels of heterogeneity of programmers and users, and interestingly, might increase in the level of forking increases (i.e. decreases in the level of coordination of the OS community). From the welfare perspective, the public policy supporting OS community to reduce its level of forking is socially profitable only if the programmers in the labor market are sufficiently heterogeneous. Finally, we extend to examine the competition in three dimensions. First, we evaluate the effectiveness of a public policy that is imposed to increase coordination. Second, the OSS

is offered by a service provider that brings higher utility from adopting the OSS. Third, the firm could contribute to the OS community to enhance the software differentiation.

Related literature. There is a large literature on the competition between proprietary firms and OS community. Bitzer and Schroder (2005) examine the innovation impact of OSS entry in the software market. In their model competition takes place on technology rather than on price or quality. They show that the evolution of the market structure from a monopoly to a duopoly exerts a positive effect on innovation. Lin (2004) proposes a model of duopolistic competition between OSS and PS in which users are heterogeneous in terms of skill and experience. She finds that the market may tip to the OSS if it provides significant benefits to users who can develop customized tools and applications using the OSS. Gaudeul (2007) studies the competition between OSS and PS when the OS community faces imperfect coordination between its programmers. She assumes that users have different valuations of software features and programmers are differentiated in term of programming costs. Gaudeul shows, that in equilibrium, the OSS will always coexist with proprietary software; low-income users and highly skilled programmers opt for an OSS, and all the others adopt a PS. Casadesus-Masanell and Ghemawat (2003) propose a mixed duopoly model to analyze the competition between Linux and Microsoft in a dynamic setting. Linux is characterized by strong demand-side effect and Windows by a larger initial installed base. The authors show that, if Microsoft keeps its initial advantage in terms of installed base and uses its market power to set its price strategically, Linux will never become the market leader. The result of the competition would be the coexistence of two or Linux software exits the market unless the cost difference between the two software increases significantly or strategic buyers such as public institutions and large corporations commit themselves to the development of Linux.

Mustonen (2003) addresses the constraints that proprietary software faces, both on the labor market of programmers and on the product market when it competes with an OS community. However, our approach differs since we consider the effects of forking. Our model is also relevant to Llanes and Elejalde (2013) on coexistence of PS and OSS, Niedermayer A. (2013) on investment incentive, Giuri et al, (2013) on the labor market; Robles et al (2012) on forking.

2 Baseline model

In the baseline model, the firm and the OS community can interact in both the labor market and software market. In the labor market, the firm sets a wage level to attract developers. Indeed, by setting the wage level, the firm determines the number of developers working for not only the PS project but also the OSS project. This would then affect the quality of each software in the product market. We will compute the optimal wage in a two stage game.

2.1 Labor market

We consider a continuum of developers of mass 1 with identical skills and abilities, but they have heterogeneous preferences with respect to the softwares. We assume that preferences are distributed uniformly along the Hotelling line [0, 1]. The PS is located at 0 while the OSS is located at 1. The location of a programmer is indexed as η, and is associated with his cost of programming software. Parameter $c > 0$ captures the unit cost of developing the software or "*mismatch*" . Here $c\eta$ is the programmer's unit coding cost of an application for the PS, and $c(1-\eta)$ is a programmer's coding cost of an application for the OSS. Each programmer develops only one application of software per unit of time which correspond to its own productivity, i.e., he chooses either to work for the firm or join the OSS community.

Let us assume that the firm pays the same wage, denoted as w, to each of its programmer while the OS community offers them the opportunity to signal their ability yielding gross utility of r, where $w > 0$ and $r > 0$. The net utility of programmer η is:

$$U_D = \begin{cases} w - c\eta \text{ if he works for the firm} \\ r - c(1-\eta) \text{ if he works for the OS community} \end{cases}$$

The parameters w and r are assumed to be sufficiently large for the entire labor market to be fully covered. This assumption essentially means that all programmers have job, which seems reasonable, in the software industry.

The marginal programmer, denoted by η^*, is indifferent between working for the firm or the OS community, we have:

$$\eta^* = \frac{c - r + w}{2c} \tag{1}$$

To this end, we assume that the quality of a software is proportional with the number of

employed developers. Furthermore, the OS community is subject to the risk of forking, i.e., the splitting of the OSS project or its development into a variety of applications which may waste labor resources of the OS community (Lerner and Tirole, 2002). In this paper, we consider that forking reduces quality of the OSS due to a lack of coordination or overlaping of programming efforts. For simplicity, we assume that the quality of the proprietary software (i.e., V_P) and the OSS (i.e., V_O) are as follows:

$$\begin{cases} V_P = \eta^* \\ V_O = \lambda(1-\eta^*) \end{cases}$$

where $0 < \lambda < 1$ represents the level of coordination or forking. Intuitively, the OS community fully coordinates if $\lambda = 1$, or forking leads to disappearance of the competing OSS if $\lambda = 0$.

2.2 Product market

There is a unit mass of users who are uniformly distributed along the Hotelling line [0,1] corresponding to their heterogenous preferences. The proprietary software is located at 0 while the OSS is located at 1. The transportation unit cost is denoted as t, and p is price of the proprietary software. The user located at x obtains the net utility, as follows:

$$U_E = \begin{cases} V_P - tx - p \text{ if he adopts the proprietary software} \\ V_O - t(1-x) \text{ if he adopts the OSS} \end{cases}$$

The marginal user located at x^* is indifferent between adopting the proprietary software or OSS as follows:

$$x^* = \frac{V_P - V_O - p + t}{2t} \tag{2}$$

2.3 Equilibrium

In the baseline model, our analysis is based on a two-stage-game. In the first stage, the firm sets the wage level to recruit developers from the labor market to work on the proprietary software. The higher wage leads to higher, but lower respectively, number of developers for the proprietary software and OSS respectively. Put it differently, the firm determines the quality of both softwares from setting wage level. In the second stage, the firm sets optimal price of its software while the OSS is offered free of charge. Users evaluate both software and decide to buy the PS or to download the free OSS.

Specifically, we assume there is no fixed or marginal costs of production. So the firm's

profit function is:

$$\Pi = (p - w)x^* \tag{3}$$

We solve the firm's optimal strategy using backward induction. In stage two, substituting (1) and (2) into (3), and then solving the F.O.C of (3), we obtain the optimal price as follows:

$$p = \frac{(w - c - r)\lambda + (2ct + 2cw + c - r + w)}{4c} \tag{4}$$

Replacing price provided in (4) into (3), and then solving for the F.O.C with respect to w to obtain the optimal wage:

$$w^B = \frac{(c + r)\lambda - (2t + 1)c + r}{1 + \lambda - 2c} \tag{5}$$

Taking the first derivative $\frac{\partial w^*}{\partial \lambda}$, it can be proved that when $c + r - t - 1 > 0$, the optimal wage is increasing as the level of coordination among OS programmers decreases, i.e., λ is close to 0. This result is counterintuitive since the firm does not always take advantage from the declining quality of the OSS due to forking to lower the wage in the labor market.

Replacing (5) into (4), (2) and (1), one can derive the equilibrium PS price (p^B), market shares in the product market (x^B) and the labor market (η^B).

3 Extension

We extend our baseline model to three cases. First, there is government intervention to enhance coordination of the OS community. Second, the OSS is provided by a supporting provider that can lead to higher utility that an user can derive from adopting the OSS. Third, the firm could contribute to the OSS development and strategically enhance the degree of differentiation between the two softwares.

3.1 Public policy to enhance coordination

In this section, we analyze the implications of public policy supporting OS community on welfare, focusing on the possibility that helping OS community to reduce its level of forking (i.e. to better coordinate development efforts of its programmers) can induce the socially optimal level of the quality software. The public policy supporting OS community can take the form of subsidizing institution of the OS movement that try to coordinate software development and standard setting (Schmidt and Schnitzer, 2003).

We can model this policy as an increase of λ. This would increase the relative attractiveness of the OSS in comparison with the PS. Since we consider that the price paid by users and the wage earned by programmers as a pure transfer that does not affect social welfare, the social welfare is the total sum of sum of firm's profit and users' net utility, minus the disutility due to the network's departure from the most ideal location, and minus the disutility of programmers. Precisely, we have:

$$W = \int (V_P - tx)dx + \int (V_O - t(1-x))dx - \int c\eta d\eta - \int c(1-\eta)d\eta - wV_P \quad (6)$$

Evaluating the above equation, we have:

$$W = (x-w)V_P - \frac{c(1-\eta)^2}{2} - \frac{tx^2}{2} - \frac{c\eta^2}{2} - \frac{(x-1)(2V_O - t - tx)}{2} \quad (7)$$

We then compare social welfare for that particular case before and after public intervention.

3.2 Investment incentive of OSS service provider

We consider that the utility of an user derived from adopting the OSS if it is distributed by a supporting provider thanks to more tailored designed, training... In this subsection, the firm competes directly with a OSS service provider. The OSS is sold at a price $l > 0$. The utility derived the OSS is increased by μ that would cost the service provider $c(\mu)$.

We then investigate competition in a three stage game as follows. First, the firm set a wage level. Second, the OSS service provider set μ. Third, competition takes place in the product market. Again, we will solve the game backwards.

3.3 The firm strategic contribution to the OSS development

The firm could strategically contribute to the OSS development by, e.g., sending some developers to work on the OSS, to learn about the OSS's strength and weakness.Then the PS and OSS will therefore become more differentiated represented by the transportation parameter in the product market. Precisely, assuming that due to the firm's strategic investment, the transportation cost from adopting the OSS (t) is increased by an amout Δt. The associated cost of the firm would be $c(\Delta t)$.

In other words, the firm can not only indirectly determine the OSS's quality, but also the degree of software differentiation. Furthermore, there is asymmetry in the transportation cost in the product market. We are interested in characterize the optimal strategy of the firm in this context.

4 Conclusion

Open source software has become a serious competitor of proprietary software not only in the product market but also in the labor market. In the labor market, the OS community creates an alternative job career for the programmers, while in the product market it offers an opportunity for users to download the OSS for free as an alternative to PS. In this context, we propose a theoretical model to study how a proprietary firm interacts with an OS community both in the labor market and in the product market under conditions of preferences heterogeneity and when the OSS development suffers from forking. In particular, we focus on the optimal price and quality of a proprietary firm. The quality is endogenous and is dependent on the number of programmers developing the PS and the firm can determine this number by paying wage.

We study also the impact on welfare of public policy supporting OS community to reduce its level of forking. Our research suggests that if the level of differentiation between OSS and PS from the users' perspective is high and that from the programmers' perspective is low, the firm will invest more in quality of its product by setting higher wage in the labor market to make it unprofitable for the programmers to join the OS community. Conversely, if the level of differentiation between OSS and PS from the programmers point of view is quite high, but less than that from the users point of view, the firm will not engage in an aggressive quality strategy and leave a large market share for the OS community.

Our paper also provides implications for public policy. It suggests that if the government can correctly evaluate the magnitude of programmers heterogeneity in the labor market, it can increase the quality of software available to users in the product market by helping the OS community to better coordinate the development efforts of its volunteers programmers and consequently decrease its level of forking. Thus, finding that programmers are relatively more heterogeneous in the labor market should be an argument for this

public intervention. Finally, our research can also analyze the strategies of the market players in the software industry.

References

[1] Baake P.,.Wichmann, T., (2003), "Open source software, competition and potential entry", Berlecon Research papers 005.

[2] Bessen, J., (2004), "Open source software: Free Provision of Complex Public Goods", http://ssrn.com/abstract=588763.

[3] Bonaccorsi, A., C. Rossi, (2003), "Why open source software can succeed, Research Policy, Special Issue on open source software development", 32(7), 1243-1258.

[4] Browne C., (1999), "The economics of free software", www.ntlug.org/~cbbrowne /freecon.html.

[5] Bitzer.J, (2004), " commercial versus open source software: the role of product heterogeneity in competition", Economic systems 28, 369-381.

[6] Bitzer J., Schröder P., (2005), "the impact of entry and competition by open source software on innovation activity", Working paper, Free university Berlin Germany and Aarhus School of Business, Denmark.

[7] Bitzer J., Schröder P, (2005), "bug- fixing and code- writing: the private provision of open source software". Information Economics and Policy 17, 389–406.

[8] Casadesus-Masanell, R., and P., Ghemawat, (2003), "Dynamic Mixed Duopoly : A Model Motivated by Linux vs. Windows", Strategy Unit Working Paper 04-012, graduate School of Business Administration, Harvard University.

[9] Dalle J-M., Juillien N (1999) Nt vs. Linux, or some explorations into the economics of free software, mimeo, ENS Cachan & IDHE-CNRS.

[10] Dasgupta P., David P. (1994) Toward a new economics in science, Research Policy 23(5), 487-521.

[11] Franke, N., E., von Hippel, (2003), "Satisfaying heterogeneous user needs via innovation toolkits: the case of Apache security software", Research Policy, Special Issue on Open Source Software, 32 (7), 1199-1215.

[12] Gaudeul, A., (2008), "Consumer welfare and market structure in a model of competition between open source and proprietary software", International Journal of Open Source Software and Processes.

[13] Gaudeul, A., (2007), "Competition between open source and proprietary software: the (La) TeX case study", Review of Network Industries

[14] Johnson, J.P., (2002), "Open Source Software: Private provision of public good", Journal of Economics and Management Strategy, 11 (4), pp. 637- 662.

[15] Johnson, J., (2004), "Collaboration, Peer Review and Open Source Software", Working Paper, http://ssrn.com/absract=535022.

[16] Johnson, J. P. (2002). Open Source Software : Private Provision of a Public Good. Journal of Economics & Management Strategy

[17] Firms' contribution to open-source software and the dominant user's skill », en collaboration

[18] Jullien N. 2009, "Firms' contribution to open-source software and the dominant user's skill", European Management Review.

[19] Kuan J., (2001), "Open source software as consummer integration into production", Haas School of Business, University of California- Berkeley.

[20] Kumar et al, (2011), Competitive Strategy for Open Source Software, Marketing Science

[21] Lakhani, K. R. and von Hippel, E. (2003). How open source software works: "free" user-to-user assistance. Research Policy, 32(6):923 { 943

[22] Lerner J., Tirole J., (2005), "The Economics of Technology Sharing: Open Source and Beyond", The Journal of Economic Perspectives.

[23] Lerner J., Tirole J., (2002), "The Simple Economics of Open Source", The Journal of Industrial Economics, volume L 0022-1821. No.2.

[24] Lerner J., Tirole J., (2001), "The Open Source Movement: Key Reseach Questions". European Economic Rewiew 45, 819-826.

[25] Lihui H., (2004), "Competition between Proprietary and Open Source Software with Heterogeneous Users", Working paper, Boston University School of Management.

[26] Gastón Llanes, Ramiro de Elejalde (2013), "Industry equilibrium with open-source and proprietary firms", International Journal of Industrial Organization

[27] Giuri et al, (2013), Skills, division of labor and performance in collective inventions: Evidence from open source software, International Journal of Industrial Organization

[28] Mustonen M., (2003), " copyleft- the economics of Linux and other open source software", Information Economics and Policy, 15 (1), pp. 99- 121.

[29] Niedermayer A. (2013), "On platforms, incomplete contracts, and open source software", International Journal of Industrial Organization

[30] O'Mahony, S. and Ferraro, F. (2007). The Emergence of Governance in an Open Source Community. Academy of Management Journal, 50(5):1079{1106.

[31] Raymond, E.S, Cathedral & the Bazaar: Measuring on Linux and Open Source by an AccidentalRevolutionary. O'Reilly, Cambridge, MA.

[32] Robles G., González-Barahona J.M. (2012), "A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes". In: Hammouda I., Lundell B., Mikkonen T., Scacchi W. (eds) Open Source Systems: Long-Term Sustainability. OSS 2012. IFIP Advances in Information and Communication Technology, vol 378. Springer, Berlin, Heidelberg

[33] Schmidt K., Schnitzer M., (2003), " Public subsidies for open source? Some economic policy issues of the software market", CERP Discussion paper

[34] Stallman, R. (1994). "Why software should not have owners". Available at http://www.gnu.org/philosophy/whyfree.html.

[35] Xing M. (2014), "The impact of commercial open source software on proprietary software producers and social welfare", Journal of Industrial Engineering and Management