Lanzi, Diego

Working Paper

Copyleft vs Copyright: some competitive effects of Open Source

Quaderni - Working Paper DSE, No. 541

Provided in Cooperation with:
University of Bologna, Department of Economics

Suggested Citation: Lanzi, Diego (2005) : Copyleft vs Copyright: some competitive effects of Open Source, Quaderni - Working Paper DSE, No. 541, Alma Mater Studiorum - Università di Bologna, Dipartimento di Scienze Economiche (DSE), Bologna, https://doi.org/10.6092/unibo/amsacta/4749

This Version is available at:
https://hdl.handle.net/10419/159382

Copyleft vs Copyright: some competitive effects of Open Source

Diego Lanzi

University of Bologna, Department of Economics, 2 Piazza Scaravilli, Bologna, Italy

June 2005
Preliminary draft

Jel Classification Codes: D23, L11, L15

KEYWORDS: COPYLEFT, OPEN SOURCE SOFTWARE, KNOWLEDGE ACCUMULATION, OLIGOPOLISTIC COMPETITION, PRODUCT DIFFERETIATION

Abstract: In this paper, we study oligopolistic competition between closed and open source softwares. By intersecting existing economic contributions on open source, we propose a two stage game with perfect information and product differetiation in which producers firstly set software's quality, then they determine prices (constrained at zero for open source programs). In doing this, we explicitly model lock-in effects, network externality components' of software quality as well as knowledge accumulation in software use and implementation.

With respect to a monopolistic benchmark case, we argue that in duopoly a proprietary sofware producer facing an open source software will reduce its selling price whether: (i) its network of users is larger than open source's one and its consumers are largely experienced on its program, (ii) it has a small network of un-skilled consumers. In opposition, after open source software's emergence, proprietary software price does augment if proprietary software users form a large, but poorly skilled network. Furthermore, we show that, in all above cases, proprietary software quality increases because of the existence of a open source alternative to a previouisly monopolistic program.

Finally, by modeling knowledge accumulation processes through difference equations, we show that the ratio between closed and open source programs' opportunity costs of software learning and deployment plays a crucial role in shaping market outcomes. Until an open source software remains too complex and technical for unskilled or time-scarse users, a *shared market* solution in which both softwares are adopted is predicted. In contrast, if opportunity costs in learning and understanding open source programs are remarkably low, or at least equal to opportunity costs of a closed source software, then a *open source dominance* outcome (i.e. all software are open ones) phases out.

1 Introduction

In the last years, Open Source Software (OSS, henceforth) has gained large market shares in several computer software markets from server applications to internet mail browers or audio editors. Surprisingly, even in the *Microsoft-dominated* market for operative systems, *Linux* is receiving considerable attention by users and software developers. The reason of this success is threefold.

First of all, OSS is freely developed by hackers and freely distributed on the web. Whether a final user has sufficiently good skills in informatics, he or she can download any needed program or utility to have a complete operative system with several applications.

Secondly, OSS can be freely modified by manipulating its source and compile codes. In this way, consumers can adapt operative system's peculiarities to what they will do with that computer and with installed applications. Moreover, any modification of the original program can be freely distributed under a *copyleft* regime[1]. Copyleft states that any modification of an open source product can be freerly distributed together with its source and compile codes *at the condition that the licence grants similar rights over new modifications.* Thus, software recipients can modify the program and, at condition that original authors are evidently quoted, can distribute their version of the product to other users under a new copyleft licence. This permits decentralized programming and networking among users.

Finally, frequent peer-to-peer exchanges of open sorce software's modifications and improvements create a *sharing community* from which free support and consulting can be exchanged.

Indeed, the "*gift and share*" philosophy of open source communities have posed some questions to economics. Among others: Why should developers work on software improvements for free ? Which are main market effects of OSS diffusion ? When will a user decide for the adoption of an OSS instead of a proprietary one ? Which effects will have the emergence of open source alternatives on proprietary software's price and quality ? And then, which effects on social welfare ?

As we review in the next section, some answers to above issues have been proposed by the so called *economics of open source*. Nevertheless, at the best of our knowledge, no papers have modeled competition between open and closed softwares when vertical product differentiation is allowed. Hence, in what follows, we firstly discuss a monopolistic case in which only a proprietary product exists, then we suggest a game in which competition between closed source and open source softwares is studied with a model of *oligopolistic competition on price and quality* (Shaked and Sutton (1982), Gabzewicz and Thisse (1979)). More precisely, we propose a two stage game with perfect information in which

[1] For a definition of *copyleft* see GNU Project (2000).

producers firstly set software's quality, then determine selling prices (obviously constrained at zero for OSS). In doing this, we will take into account lock-in effects, network externality components' of software quality as well as knowledge accumulation in software use, all crucial elements for shaping competition in software markets (see Shy (2001)).

Our results show that in duopoly a proprietary sofware house, which faces an open source software, reduces its selling price with respect to the monopolistic case if: *(i)* its network of users is larger than open source's one and its consumers are largely experienced on its program; *(ii)* its network of users is small and largely un-skilled. In opposition, after open source software's emergence, proprietary software price does augment if proprietary software users form a large, but poorly skilled, network. Furthermore, we show that, in all above cases, proprietary software quality increases because of the existence of a open source alternative to a previouisly monopolistic program.

Finally, by modeling knowledge accumulation processes through difference equations, we show that the ratio between closed and open source programs' opportunity costs of software learning and deployment shapes market outcomes. More precisesly, until open source softwares remain too complex and technical for unskilled or time-scarse users, a *shared market* solution in which both softwares are adopted can be predicted. In contrast, whereas opportunity costs in learning and understanding open source programs are remarkably low, or at least equal to opportunity costs of closed source ones, then a *open source dominace* solution (i.e. all software are *open* ones) phases out.

The analysis is organized as follows. In Section 2, in order to motivate our set up,.we briefly review existing economic literature on OSS. In Section 3, we present our model together with basic definitions and notation. Price and quality competition are analyzed in Section 4 in the case of exogeneously fixed users' knowledge and effort in software learning. Such an assumption is removed in Section 5 where knowledge accumulation dynamics with endogeneous effort in software learning and deployment are analyzed. As ususal, the last paragraph briefly concludes.

2 Economic Literature on Open Source

Economic analysis of OSS is relatively young. Its birth can be linked with the publication of a short article in which Josh Lerner and Jean Tirole suggest some research questions that the open source movement raises to economists (Lerner and Tirole (2001)). So far, answers to these questions have been only partially found by economic analysis. In this section, we present a selection of these contributions[2].

[2] For a review on early results of the economics of open source see Schiff (2002). A more recent survey on Open Source is Lerner and Tirole (2004).

In a seminal paper, Lerner and Tirole (2002) discuss programmers' economic incentives to participate in OSS projects[3]. Through a *descriptive approach* (i.e. some interviews with leaders of the OS movement), they are able to identify immediate and delayed benefits of participation. Lerner and Tirole classify them as follows: (a) monetary compensations paid by programmers' employees for experience accumulation in software manipulation, (b) software customization and bugs fixing, (c) ego gratification through peer recognition, and, (d) signalling purposes within programmers' communities for career concerns. As Lerner and Tirole argue, these expected payoffs explain why sophisticated users are likely to contribute to OSS projects. Because of their relatively more advanced skills and lower opportunity costs of time, skilled programmers or heavy and educated users receive large net benefits from participation and open source software adoption. The same cannot be said for un-skilled and light users.

A different approach to explain participation is followed by Johnson (2001). By using a game-theoretic set-up with n players, he models programmer choice (participate vs not-participate) in the case of exogeneously fixed benefits and costs of participation and some beliefs on other agents' behaviour. His results show that in a Bayes-Nash equilibrium a programmer participates to OSS projects only if he/she *benefit-cost ratio* is larger than a threshold that increases with respect to the probability that a software development will be implemented by another user. By increasing n, the likelihood that an agent with large benefit-cost ratio exist augments as well as incentives to *free-ride* increase. Without particular assumptions on beliefs distribution, Johnson shows that expected social welfare is increasing in n, i.e. larger OSS communities provide larger benefits to their members (the so called *size-effect*).

Van Wegberg and Berends (2000) focus on software adoption by consumers with different experiences and skills in software utilization and deployment. Software users maximize utility with respect to effort in software development and to the quantity of feedbacks (to other developers of open source programs or to the proprietary software house). Any consumer can buy for an OS program or a proprietary one and, in both cases, an higher level of effort increases users' experience and utility. Moreover, whether a user adopts an open source program his/her feedbacks create reputation within the OS community and more dense peer-to-peer exchanges with larger communites increase user's utility. On the contrary, by adopting a closed source product, a consumer enlarges software house's network and profits and thus its investment in software quality.

By simulating model's first order conditions, Van Wegberg and Berends (2000) show that: *(a)* skilled users opt for an OSS immediately after its release; *(b)* an increase in highly-qualified pioneers augments the long-run number of

[3] In this section, we do not discuss contributions in which social, collective or non-economic motivations explain programmers' behaviour. On these issues see Hertel et. al (2003), Bonaccorsi and Rossi (2003) and Zitlyn (2003).

OSS users and overall OSS diffusion. If a sufficiently large numbers of highly-skilled programmers join to the OS community, low taleted users will adopt OSS as well and the OSS will entirely dominate the market (*open source dominance solution*). In opposition, whether initial adoption of the OSS is sticky and only few pioneers participate in OS projects, larger network effects motivate the adoption of a proprietary product. In such a case, a *shared market solution* emerges.

Competition between open and closed source softwares has been modeled by Mustonen (2003), Scmidt and Schmitzer (2003), Lin (2004) and Gandel (2004).

Gandel (2004) analyzes duopolistic competition between an OSS and a proprietary one when the OSS soffers from costs of mis-coordination in development (for instance, *forking* in code writing) and proprietary software's distribution costs are different from zero. Consumers are differentiated not only in terms of skills and experience, but also with respect to willingness to pay. In Gandel's framework, a *shared market* solution is always reached: low-income consumers and well endowed-highly skilled users adopt an OSS, all others opt for a proprietary product.

From a different viewpoint, Mustonen (2003) develops a two stage model in which a monopolist with a copyright software firstly invests in his program quality by hiring programmers, then it determines software selling price. The existance of a OSS constrains monopoly firm's behaviour becuase consumers may decide for a free software and programmers for participation in OSS projects. Furthermore, users face non-negative *implementation costs*, i.e. costs of installing and learing the software. Mustonen's results show that whether implementation costs are remarkably low some consumers prefer an OSS reducing monopolist's price and profit. In opposition, whether these costs are high, the monopoly firm continues to set a monopolistic price for its product[4].

Similar conclusions are drawn by Lin (2004). By modeling duopolistic competition between an OSS and a proprietary one in presence of heterogeneous consumers (in terms of skills and experience), implementation costs and network effects, she argues that if OSS is simply a cost-saving alternative to the proprietary software, then the latter either shares the market with an OSS or it becomes the unique adopted program. More precisely, whether agents are highly skilled and experienced, many consumers adopt an OSS and the proprietary software house reacts by reducing its selling price to gain market share. Such a price reduction will be so strong to cover the whole market[5]. In oppo-

[4] There is also an intemediate case in which the proprietary software firm sets the minimun price for detering the marginal consumers to switch to a copyleft program. For details, see Mustonen (2003).

[5] Casadesus-Masanell and Ghemawat (2003) models competition between an OSS and a proprietary one in presence of network effects. Their results too indicate that the mergence of an OSS lowers proprietary software price.

sition, if agents are poorly skilled and experienced, few users choose an OSS and the proprietary software house sets an higher price leaving market share to its open source alternative. The only case in which the market tips to the OSS is when it ensures better performances and provides customized benefits to users. Whether these benefits are very large, the proprietary software firm will continue to loose market share even if it reduces its selling price. If this is the case, we get a *dominance solution* in which all existing softwares are open source ones[6].

Finally, Schmidt and Schmitzer (2003) propose an Hotelling-inspired model of spatial competition between a proprietary software and an OSS. In their set-up, a group of user is locked-in with an OSS, another group is locked-in with a proprietary program and only a third subset of consumers can choose between a free OSS or a proprietary one. Programs are completely compatible (hence network effects ignored) and users face *trasportation costs*, i.e. costs of adaptation of the program to their operative system's specificities. As Schmidt and Schmitzer show, in equilibrium an larger number of OSS users (for instance, obtained through public subsidies to open source) reduces proprietary software house's profits and investment in R&D as well as an it increases proprietary software price for locked-in consumers.

This last contribution can introduce us to our aims. In Schmidt and Schmitzer (2003)'s paper an increase in the number of OSS's users augments proprietary software price and reduces firm's profit and incentives to innovate. This is caused by the fact that the proprietary software house will focus more on the group of consumers that cannot adopt an OSS (i.e. locked-in users) than on the set of undecided users. By increasing its selling price for locked-in consumers, the proprietary software house tries to balance the reduced demand for its product caused by the emergence of an OSS.

Hence, it seems that two *price effects* can follow the introduction of OSS in the software market: a proprietary software's price cut to maintain proprietary software house's market share and a price increase to get larger profits from locked-in users. Nevertheless, in the existing literature, these two effects cannot co-exist in the same analytical set-up. Thus, our modest attempt is to provide a set-up in which both effects can be rationalized. Furthermore, we show that the likelihood of an OS dominance solution depends on the rate of experience accumulation for OSS with respect to proprietary ones. Accordingly to Mustonen (2003), we argue that low implementation costs are crucial for OSS's dominance.

[6] On convenience to release programs as OSS, Hawkins (2004) argues that whether technological spillovers (i.e. savings in software development costs by competitors R&D activites) and related commercial activities' (i.e. assistance or internet based connection with dedicated portals) revenues are high, competitive firms' dominant strategy is to release its product under a copyleft licence. These can also be new competitive dimensions in software markets with skilled users and customized OSS.

3 The Model

In this section we introduce definitions and notation. The set up proposed is inspired to a well established tradition of models of duopolistic competition with vertically differentiated products and perfect information (Shaked and Sutton (1982), Gabzewicz and Thisse (1979)) The main difference with those models is that we consider a profit-oriented firm (*the software house*) and a no-profit producer (*the OS community*). Hereafter, we assume as exogeneously fixed users' effort in software learning and implementation. We shall drop this assumption in Section 5.

3.1 The Demand Side

Let us suppose to have N undecided consumers equal in all respects except for individual ability in software implementation, use and deployment (h). For the sake of simplicity, we assume that individual abilities are uniformly distributed on $\left[\underline{h};\overline{h}\right]$ with $\overline{h} = \underline{h} + 1$. Any user can buy only a unit of product and he/she can choose between a proprietary software (i) or an OSS (j). Let us assume that $\underline{h}$ is sufficiently high to guarantee that even the least skilled user adopts a software.

Consumers' preferences are represented by the following utility function:

$$u_s^t = q_s^t h^t K_s^t - p_s^t - \omega_s^t e_s^t \qquad \text{for } s = i, j \tag{1}$$

where q_s^t is the software quality, K_s^t is consumer's experience in software use and deployment, e_s^t indicates users' effort in software implementation, ω_s^t the monetary opportunity cost of such an effort, p_s^t is the price of software and t denotes a time period. Obviously, $p_j^t = 0$.

As in Van Wegber and Berends (2000), users' experience is increased by effort as well as decreased by knowledge dispersion and obsolescence. The following difference equation characterizes the experience/knowledge accumulation process:

$$K_s^t = (1 - \gamma) K_s^{t-1} + \beta_s e_s^t \tag{2}$$

with $s = i, j$, $\gamma \in [0; 1]$ and $\beta_s \in \left\{\underline{\beta};\overline{\beta}\right\}$. γ is the knowledge dispersion rate and β_s indicates the marginal productivity of effort.

Consistenly with the view of more active and dense feedbacks within OSS's communities than between proprietary software's consumers and producers[7], we assume that $\beta_i = \underline{\beta} < 1$ and $\beta_j = \overline{\beta} < 1$ with $\underline{\beta} < \overline{\beta}$. Moreover, higher technicality of OSS motivates the assumption that $\omega_j^t > \omega_i^t$.

As in Schmidt and Schmitzer (2003), accumulated knowledge in software use and implementation can be only imperfectly transferred between softwares such that *swithcing costs* are not equal to zero. In particular, we suppose that:

$$\begin{aligned} K_i^t &= \lambda K_j^t \\ K_j^t &= \lambda K_i^t \end{aligned} \tag{3}$$

where $\lambda \in [0;1)$ is a symmetric *lock-in coefficient*.

Software quality is defined by a network externality component and an hedonic component determined by using software's average down-time and its number of bugs. Formally, we write:

$$q_s^t = \Upsilon\left(b_s^t\right) + \delta_s^t \quad \text{for } s = i, j \tag{4}$$

where $b_s^t \in R_+$ is an exogeneously fixed number of software users (*installed base*, henceforth), $\Upsilon : R_+ \to [0;1]$ a *fuzzy* function with $\Upsilon\left(b_s < \underline{M}\right) = 0$ and $\Upsilon\left(b_s \geq \overline{M}\right) = 1$. As usual in fuzzy mathematics[8], zero stands for non-achievement of network externalities and one for full achievement[9].

Finally, $\delta_s^t \in \left[\underline{\delta}; \overline{\delta}\right]$ is an index of avarage software quality. Its upper and lower bounds are given by the maximum quality attainable with existing technologies and the minimum quality standard required by laws and regulators.

3.2 The Supply Side

On the supply side, we consider two players. On the one hand, a profit-oriented software house (i) that produces a proprietary software sold at a positive price $p_i^t > 0$. Production costs are given by a constant marginal cost of distribution, $c > 0$, zero replication costs and totally paid-off fixed costs of

[7] On this issue see Lakhani and Von Hippel (2003).

[8] See Zadek (1965).

[9] Note that full exploitation of the network component of quality is ensured if $b_s \geq \overline{M}$ and null expoitation if $b_s < \underline{M}$. Both $\underline{M}$ and $\overline{M}$ are critical bounds in network's size. By assuming that $\Upsilon' > 0$ in $\left[\underline{M}; \overline{M}\right]$, we roughtly approssimate a logistic curve.

software creation. Indeed, this cost structure violates some traditional assumptions on *costs-subadditivity* in software markets (Shy (2001)). Nevertheless, it is particularly meaningfull whether we want to analyze effects of an OSS in previously monopolistic markets dominated by well established (and well paid-off) softwares.

Let us denote with $D_i^t = D_i\left(p_i^t; q_i^t; q_j^t\right)$ the demand for proprietary software with $\frac{\partial D_i^t}{\partial p_i^t} < 0$, $\frac{\partial D_i^t}{\partial q_i^t} > 0$ and $\frac{\partial D_i^t}{\partial q_j^t} < 0$. Thus, firm i's decisional problem is given by:

$$\max_{p_i^t;\delta_i^t} \Pi_i^t = D_i^t\left(p_i^t - c\right) \tag{5}$$

On the other hand, we consider an OSS's community that freely develops and distributes its software (i.e. $p_j^t = c_j = 0$) and faces a demand equal to $D_j^t = D_j\left(p_i^t; q_i^t; q_j^t\right)$ with $\frac{\partial D_j^t}{\partial p_i^t} > 0$, $\frac{\partial D_j^t}{\partial q_i^t} < 0$, $\frac{\partial D_j^t}{\partial q_j^t} > 0$ and $D_j^t + D_i^t = N$. Let us suppose that the OSS' community objective is to include the largest share of the population of users. Consistently, we describe player j's decision problem as follows:

$$\max_{\delta_j^t} D_j^t = N - D_i^t \tag{6}$$

Finally, let us suppose that quality and price decisions are assumed sequentially and that the interaction between players has the form of a a two stage game with perfect and complete information. In the first stage of the game, players decide products' quality, then, in the latter stage, firm i decides its selling price. As usual, we use a backward induction procedure and derive subgame perfect Nash equilibria.

4 Some Competitive Effects of Open Source

In this section, in order to analyze how the introduction of an OSS changes price and quality decisions of a proprietary software house we compare results in the case of oligopolistic competition with a benchmark *monopoly case*.

4.1 A Monopolistic Software House

Suppose that no OSS exists. Users have to decide whether or not to but a proprietary software. Obviously, only consumers with $u_i^t \geq 0$ will buy the program. In this case, firm i's demand is equal to:

$$D_i^t = N\left(\overline{h} - \widehat{h}\right) \tag{7}$$

with

$$\widehat{h} = \frac{p_i^t + \omega_i^t e_i^t}{q_i^t K_i^t}$$

and monopoly profits $\left(\Pi_i^M\right)$ are given by

$$\Pi_i^M = N\left(1 - \frac{p_i^t + \omega_i^t e_i^t}{q_i^t K_i^t} + \underline{h}\right)\left(p_i^t - c\right) \tag{8}$$

By maximizing (8) with respect to p_i^t, it is straightforward to notice that:

$$p_i^M = \frac{1}{2}\left[\left(1 + \underline{h}\right) q_i^t K_i^t - \frac{\omega_i^t H_i^t}{\underline{\beta}} + c\right] \tag{9}$$

with $H_i^t = \triangle K_i + \gamma K_i^t$ and $e_i^t = \dfrac{H_i^t}{\underline{\beta}}$.

As it can be pointed out by looking at expression (9), the monopoly price increases with respect to software quality and costs as well as with respect to users' assumulated experience in software use and implementation (K_i^t). Furthermore, p_i^M decreases with respect to users' opportunity costs and in correspondence to an increase in the degree of knowledge dispersion (γK_i^t) or knowledge accumulation $(\triangle K_i)$. In other words, a monopolistic software house charges an high price for high quality/well-known software for which experience and knowledge accumulation is slow and a huge stock of accumulated experience characterizes users. On the contrary, the monopolist sets a low price for new (K_i^t small), low-quality products for which knowledge accumulation and dispersion rates are high.

Whether we use (9) and (4) in (8), we get profit in the quality stage of the game, these are given by:

$$\Pi_i^M = N\left\{\overline{h} - \frac{1}{\left(\Upsilon(b_i^t)+\delta_i^t\right)K_i^t}\left[\frac{1}{2}\left(\left(1+\underline{h}\right)\left(b_i^t+\delta_i^t\right)K_i^t - \frac{\omega_i^t H_i^t}{\underline{\beta}} + c\right) + \omega_i^t \frac{H_i^t}{\underline{\beta}}\right]\right\}$$
$$\left[\frac{1}{2}\left(\overline{h}\left(b_i^t+\delta_i^t\right)K_i^t - \frac{\omega_i^t H_i^t}{\underline{\beta}} + c\right) - c\right] \quad (10)$$

By looking at (10), it can be easily checked that conditions for an interior solution hold. Therefore, by maximizing (10) with respect to δ_i^t, it can be shown that equilibrium quality is equal to:

$$\delta_i^M + \Upsilon(b_i) = \frac{\left[\left(\omega_i H_i + c\right)\left(\omega_i H_i + \underline{\beta} c\right)\right]^{1/2}}{\underline{\beta} K_i \left(\underline{h}+1\right)} < \overline{\delta} + \Upsilon(b_i) \quad (11)$$

4.2 Duopolistic Competition

Next, let us consider the case in which a proprietary software compete with a freely developed and distributed OSS. Now, a user will prefer an OSS if and only if $u_j^t \geq u_i^t$ or if

$$h^t \geq \frac{\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t - p_i^t}{\left(q_j^t - q_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}} = \widetilde{h} \quad (12)$$

with $H_s^t = \triangle K_s + \gamma K_s^t$ and $e_s^t = \dfrac{H_s^t}{\beta_s}$ for $s = i, j$. Consistently with Van Wegberg and Bedenrs (2000) and Lin (2004), expression (12) states that highly skilled users adopt an OSS. In this case, players' objective funtions are given by:

$$\Pi_i^t = N\left[\widetilde{h} - \underline{h}\right]\left(p_i^t - c\right) = N\left[\frac{\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t - p_i^t}{\left(q_j^t - q_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}} - \underline{h}\right]\left(p_i^t - c\right) \quad (13)$$

$$D_j^t = N\left[\overline{h} - \widetilde{h}\right] = N\left[\overline{h} - \frac{\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t - p_i^t}{\left(q_j^t - q_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}}\right] \quad (14)$$

In the second stage of the game, firm i sets its best reply to a null price of an OSS. Hence. by maximizing (13) with respect to p_i^t, we can determine the optimal proprietary software price given by:

$$p_i^* = \frac{1}{2}\left[\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t + c - \underline{h}\left(\Upsilon(b_j^t) + \delta_j^t - \Upsilon(b_i^t) + \delta_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}\right] \tag{15}$$

We are equipped to show the following:

Proposition 1 *In the subgame perfect Nash equilibrium, both players implements.the maximum level of software hedonic quality.*

Proof. By substituting (15) in (13) and (14) we obtain first stage payoffs. These are given by:

$$\begin{aligned}\Pi_i^t &= N\left[\frac{\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t - \frac{1}{2}\left[\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t + c - \underline{h}\left(\Upsilon(b_j^t)+\delta_j^t - \Upsilon(b_i^t)+\delta_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}\right]}{\left(\Upsilon(b_j^t)+\delta_j^t - \Upsilon(b_i^t)+\delta_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}} - \underline{h}\right] \\ &\left(\frac{1}{2}\left[\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t + c - \underline{h}\left(\Upsilon(b_j^t) + \delta_j^t - \Upsilon(b_i^t) + \delta_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}\right] - c\right)\end{aligned} \tag{16}$$

and

$$D_j^t = N\left[\overline{h} - \frac{\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t - \frac{1}{2}\left[\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t + c - \underline{h}\left(\Upsilon(b_j^t)+\delta_j^t - \Upsilon(b_i^t)+\delta_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}\right]}{\left(\Upsilon(b_j^t)+\delta_j^t - \Upsilon(b_i^t)+\delta_i^t\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}}\right] \tag{17}$$

Given that $\dfrac{\partial \Pi_i^t}{\partial \delta_i^t} > 0$ and $\dfrac{\partial \Pi_j^t}{\partial \delta_j^t} > 0$, it does not exist an interior Nash equilibrium in the second stage of the game. A corner solution, in which both players implement the maximum level of product hedonic quality, holds i.e. $\left(\delta_i^*; \delta_j^*\right) = \left(\overline{\delta}; \overline{\delta}\right)$. ■

By comparing Proposition 1's insights with (11), we are able to stress a first effect of the introduction of OSS in software markets. In order to attract in the community the largest number of users, the OS developers release an highest quality program. Hence, the proprietary software house increases its product quality (with respect to the monopoly case) in order to not loose market share. Quality imitation at the highest level leads to a *minimum differentiation result*.

Furthermore, the existence of a freely distributed and developed program produces different effects on software i's price in correspondence with different levels of experience accumulation and with respect to the relative number of users of each software. Let us firstly deal with the more realistic case of $\Upsilon(b_i^t) - \Upsilon(b_j^t) \geq 0$.

With a larger installed base for the proprietary software than for the OS one, firm i's second stage profits are non-negative only if proprietary software's users have large experience in software use and implementation compared with users of OSS (*i.e.* $K_i^t - K_j^t > 0$). Whether the experience accumulation

advantage of proprietary software is sufficiently large, lock-in effect are strong. Therefore, the proprietary software house augments its selling price on these users to compensate its reduced market share. On the contrary, whether the difference in knowledge accumulation between the two programs is small, some users could switch to OSS and firm i decides for a price cut. Last statements are formally proved in the next proposition.

Proposition 2 *With* $\Upsilon(b_i^t)-\Upsilon(b_j^t) \geq 0$, *in the subgame perfect Nash equilibrium a proprietary software price is lower (resp. higher) than in the monopoly case if and only if* $K_i^t - K_j^t \geq \kappa > 0$ *(resp* $K_i^t - K_j^t < \kappa$*).*

Proof. In the subgame perfect Nash equilibrium, the price of the proprietary software is equal to:

$$p_i^* = \frac{1}{2}\left[\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t + c - \underline{h}\left(\Upsilon(b_j^t) - \Upsilon(b_i^t)\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}\right] \tag{18}$$

By comparing (18) with (9) it is immediate to show that:

$$p_i^* \leq p_i^M \Leftrightarrow K_i^t - K_j^t \geq \frac{\left[\left(\omega_i H_i + c\right)\left(\omega_i H_i + \underline{\beta}c\right)\right]^{1/2} + \underline{\beta}^2\omega_j^t H_j^t - \underline{\beta}\overline{\beta}\omega_i^t H_i^t}{\underline{h}(\Upsilon(b_j^t) - \Upsilon(b_i^t))\underline{\beta}} = \kappa > 0 \tag{19}$$

Otherwise, the opposite holds. ■

With $\Upsilon(b_i^t) - \Upsilon(b_j^t) < 0$, firm i's second stage profits are non-negative only if $K_i^t - K_j^t < 0$, i.e. buyers of proprietary software are few poorly skilled and experienced users for which an OSS is too technical and complex. In such a case, the proprietary software house always reduces its product price with respect to the monopoly case. Formally,

Proposition 3 *With* $\Upsilon(b_i^t)-\Upsilon(b_j^t) < 0$, *in the subgame perfect Nash equilibrium a proprietary software price is always lower than in the monopoly case.*

Proof. As for Proposition 2, we have to compare (18) and (9). In this case, we have that:

$$p_i^* \leq p_i^M \Leftrightarrow K_i^t - K_j^t < \frac{\left[\left(\omega_i H_i + c\right)\left(\omega_i H_i + \underline{\beta}c\right)\right]^{1/2} - \underline{\beta}^2\omega_j^t H_j^t - \underline{\beta}\overline{\beta}\omega_i^t H_i^t}{\underline{h}(\Upsilon(b_j^t) - \Upsilon(b_i^t))\underline{\beta}} > 0 \tag{20}$$

Expression (20) always holds given that $K_i^t - K_j^t$ has to be negative to avoid a degenerate solution. ■

Whether we suppose that $\Upsilon(b_i^t) - \Upsilon(b_j^t) < 0$ and $K_i^t - K_j^t \geq 0$ or $\Upsilon(b_i^t) - \Upsilon(b_j^t) \geq 0$ and $K_i^t - K_j^t < 0$, some other solutions emerge. Under

these specifications of our parameters, software house's equilibrium profits are negative for any selling price. In the first case, larger network effects within the OS community attract well-skilled and experienced consumers of proprietary softwares which accept a certain amount of knowledge dispersion to benefit of larger network effects. In the second one, large experience accumulation in OSS makes the market for a proprietary software excessively thin for profitable production. In both scenarios, the optimal choice for a proprietary software house is to switch to an OS module for its product or to exit.

Finally, with some comparative statics it is easy to check how proprietary software's price changes with different knowledge dispersion rates or lock-in coefficients. For the sake of tractability, let us assume that $\underline{\beta} = \overline{\beta} = \widehat{\beta}$. By differentiating (18) with respect to γ and λ, it can be shown that:

$$\frac{\partial p_i^*}{\partial \gamma} = \frac{\widehat{\beta}}{2}\left(\omega_j^t - \lambda\omega_i^t\right) K_j^{t-1} > 0 \tag{21}$$

$$\frac{\partial p_i^*}{\partial \lambda} = -\frac{\widehat{\beta}\omega_i^t}{2}\left[K_j^t - (1-\gamma)K_j^{t-1}\right] < 0 \;\Leftrightarrow K_j^t - (1-\gamma)K_j^{t-1} \geq 0 \tag{22}$$

Expressions (21) points out that p_i^* always increases with high knowledge dispersion and particularly skittish experience accumulation in proprietary software use and implementation. Larger values of γ reinforce lock-in effect and this allows firm i to charge an higher price. This intuition is refined by expression (22): if the lock-in coefficient diminishes, accumulated knolwdge in one software can be transferred on the competing one only by incurring in large losses. Consistently, lock-in increases when users are accumulating experience through software utilization $\left(\text{i.e. } K_j^t - (1-\gamma)K_j^{t-1} \geq 0\right)$ [10] while it diminishes if knowledge and experience in software implementation and use decreases in time.

5 Knowledge Accumulation Paths and OSS Dominance

OS softwares are new comers in the market and their relative diffusion is not large. Since, in our set up, OSS dominance is predicted for the cases of larger installed base of users ($\Upsilon(b_i^t) - \Upsilon(b_j^t) < 0$ and $K_i^t - K_j^t \geq 0$) and larger accumulated knowledge in software deployment ($\Upsilon(b_i^t) - \Upsilon(b_j^t) \geq 0$ and $K_i^t - K_j^t < 0$), it is not meaningless to study how dynamics of knowledge

[10] Note that by using (3) it can be shown that $K_j^t - (1-\gamma)K_j^{t-1} \geq 0$ implies $K_i^t - (1-\gamma)K_i^{t-1} \geq 0$.

accumulation are shaped. In order to perform this task, we need to analyze time paths of knowledge accumulation by using (2) and by endogenizing agents' effort in software use and implemetation. By studying difference equations with endogeneous effort, we shall assess that OSS may dominate the market only with low opportunity costs of OSS learning and use are assured through low implementation costs. In all other cases, the OSS is a *niche alternative* to proprietary softwares for well-skilled users.

5.1 Endogeneous Effort

In our set up, users' decision problem can be described as a constrained utility maximization problem given by:

$$\max_{e_s} u^t = q_s^t h^t K_s^t - p_s^t - \omega_s^t e_s^t \tag{23}$$

$$\text{s.t.} \qquad p_s^* + \omega_s^t e_s^t = I \tag{24}$$

$$\text{and} \qquad K_s^t = (1-\gamma) K_s^{t-1} + \beta_s e_s^t \tag{25}$$

where I is income and $s = i, j$. By substituing (24) and (25) in (23) and by taking the first order condition, we get that:

$$e_s^* = \frac{I - p_s^*}{\omega_s} \tag{26}$$

with $e_i^* = \dfrac{I - p_i^*}{\omega_i^t}$, $e_j^* = \dfrac{I}{\omega_j^t}$ and $\omega_j^t > \omega_i^t$. As it is straightforward to notice, $e_j^* \geq e_i^*$ if $\dfrac{I}{\omega_j^t} \geq \dfrac{I - p_i^*}{\omega_i^t}$ or, equivalently, if $\dfrac{\omega_j}{\omega_i} \leq \dfrac{I}{I - p_i^*}$. More explicitly, endogeneous effort in OSS deployment and implemetation will be stronger than effort spent on a proprietary product only if the *opportunity cost ratio* between softwares $\left(\psi = \dfrac{\omega_j^t}{\omega_i^t}, \text{ henceforth}\right)$ is lower than a threshold which decreases with respect to users' income and increases with respect to proprietary software's price. Thus, OSS induces stronger effort in software use and implementation within populations of poorly endowed users who pay wide price margins to software houses. In opposition, wealthy consumers in competitive software markets deeply suffer high OSS use opportunity costs. In this case, consumers are likely to prefer a more simple proprietary product.

By substituing (26) in (2), we have a complete specification of knowledge/experience accumulation euqations for both software. These are, respectively for an OSS and a proprietary one, given by:

$$K_j^t - K_j^{t-1} = \overline{\beta}\frac{I}{\omega_j} - \gamma K_j^{t-1} \tag{27}$$

$$\begin{aligned} K_i^t - K_i^{t-1} &= \underline{\beta}\frac{I - p_i^*}{\omega_i} - \gamma K_i^{t-1} \\ &= \underline{\beta}\frac{I - \frac{1}{2}\left[\underline{\beta}\omega_j^t H_j^t - \overline{\beta}\omega_i^t H_i^t + c - \underline{h}\left(\Upsilon(b_j^t) - \Upsilon(b_i^t)\right)\left(K_j^t - K_i^t\right)\underline{\beta}\overline{\beta}\right]}{\omega_i} - \gamma K_i^{t-1} \end{aligned} \tag{28}$$

In the next section, with no loss of generality, we deal with the simple case of $\Upsilon(b_j^t) - \Upsilon(b_i^t) = 0$. In doing this, we consider two separate scenarios. A first case in which no knowledge/experience dispersion takes place and a more general one where dispersion of accumulated knowledge is allowed.

5.2 Knowledge Accumulation without Dispersion

In order to analyze knowledge accumulation paths, we have to manipulate expressions (27) and (28). By recalling that $H_s^t = \triangle K_s + \gamma K_s^t$, it can be shown that above difference equations can be written as:

$$K_j^t + (\gamma - 1)K_j^{t-1} = \overline{\beta}\frac{I}{\omega_j} \tag{29}$$

$$K_i^t + (\gamma - 1)K_i^{t-1} = \frac{\Lambda}{\omega_i} - \Delta\frac{\overline{\beta}I}{\omega_i} \tag{30}$$

with $\Lambda = \frac{4\underline{\beta}I + c}{2(2 - \underline{\beta}\overline{\beta})}$ and $\Delta = \frac{\underline{\beta}^2}{2 - \underline{\beta}\overline{\beta}}$. Whether $\gamma = 0$, (29) and (30) are first order difference equations of the form:

$$y_{t+1} + ay_t = k \tag{31}$$

with $a = -1$ and $k \neq 0$[11]. Their general solutions are given by the sum of two components: a *particular integral* (that is, any solution of (31)) and a *complementary function* (i.e. the solution of the *reduced* version of (31) with $k = 0$). In our case, general solutions of expressions (29) and (30) are given by:

$$K_i^t = K_i^0 + \left(\frac{\Lambda}{\omega_i} - \Delta\frac{\overline{\beta}I}{\omega_i}\right)t \tag{32}$$

[11] On first order difference equations see Chiang (1994) pp. 502-530.

$$K_j^t = K_j^0 + \left(\overline{\beta}\frac{I}{\omega_j}\right)t \tag{33}$$

where $K_s^0 \in R_+$ indicates initial conditions for $s = i, j$. As above, the novelty of OSS allows us to suppose that $K_i^0 \geq K_j^0$, i.e. initial accumulated knowledge for an OSS is lower than that for a proprietary products. Then, by using (32) and (33), we get the following:

Proposition 4 *OSS dominates in software markets only if* $\psi < \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I}$. *In opposition, whether* $\psi \geq \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I}$ *a shared market solution is reached.*

Proof. Compare expressions (32) and (33)'s slope. Straightforwardly, we have:

$$\frac{\Lambda}{\omega_i} - \Delta\frac{\overline{\beta}I}{\omega_i} \geq \overline{\beta}\frac{I}{\omega_j} \;\Leftrightarrow\; \frac{\omega_j^t}{\omega_i^t}(=\psi) \geq \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I} \tag{34}$$

In this case, given that $K_i^0 \geq K_j^0$, accumulation paths for an OSS will never lead to larger stocks of knowledge with respect to

those for proprietary products. Hence, by recalling above discussion, we can conclude that a proprietary software will be not

driven out from the market. On the contrary, whether $\psi < \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I}$, knowledge accumulation paths for an OSS are steeper that

those referred to a proprietary product. If this is the case, identical initial conditions ($K_i^0 = K_j^0$) do involve immediate OSS

dominance since $K_j^t > K_i^t$ for any $t \neq 0$. Otherwise, if $K_i^0 > K_j^0$, $K_j^t > K_i^t$ only if $t > t^* = \dfrac{K_i^0 - K_j^0}{\left[\overline{\beta}\frac{I}{\omega_j} + \Delta\frac{\overline{\beta}I}{\omega_i} - \frac{\Lambda}{\omega_i}\right]}$. ■

Open source dominance is hence predicted only in the case that the opportunity costs ratio between software is sufficiently low, that is either the OSS is relatively simpler than the proprietary one or peer-to-peer reviews for OSS are sufficiently strong to compensate its higher technicality. With low OSS implementation costs or huge knowledge sharing among users, knowledge accumulation paths in OSS use and deployment are steeper than their proprietary couterparts. If this is the case, with $K_i^0 > K_j^0$ OSS will dominate the market after a certain lapse of time (t^*). In opposition, with $K_i^0 = K_j^0$, OSS dominance is predicted immediately after its release. As we will discuss in the next paragraph, these conclusions also hold if knowledge dispersion is allowed (i.e. $\gamma \neq 0$).

5.3 Knowledge Accumulation with Dispersion

Let us conclude by assuming that $\gamma \neq 0$. For $K_i^0 = K_j^0 = 0$, expressions (29) and (30) are first order difference equations like (31) with $a \neq -1$ and $k \neq 0$. Thus, general solutions of exponential form are given by:

$$K_i^t = -\frac{1}{\gamma}\left(\frac{\Lambda}{\omega_i} - \Delta\frac{\overline{\beta}I}{\omega_i}\right)(1-\gamma)^t + \frac{1}{\gamma}\left(\frac{\Lambda}{\omega_i} - \Delta\frac{\overline{\beta}I}{\omega_i}\right) \tag{35}$$

$$K_j^t = \left(-\frac{\overline{\beta}}{\gamma}\frac{I}{\omega_j}\right)(1-\gamma)^t + \frac{\overline{\beta}}{\gamma}\frac{I}{\omega_j} \tag{36}$$

Exactly as above, OSS dominance is predicted immediately after its release if $K_j^t > K_i^t$ for any $t \neq 0$, condition verified if $\psi < \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I}$. In opposition, whether $\psi \geq \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I}$, the two softwares share the market.

Finally, if $K_i^0 > K_j^0$ and $\psi < \frac{\overline{\beta}I}{\Lambda - \Delta\overline{\beta}I}$, an OSS dominance solution is reached after

$$t^* = \ln_{1-\gamma}\left[1 + \frac{\left(K_i^0 - K_j^0\right)\omega_i\omega_j\gamma}{\overline{\beta}I\omega_i - \omega_j\left(\Lambda - \Delta\overline{\beta}I\right)}\right] \tag{37}$$

periods of time.

6 Concluding Remarks

In this paper, competitive effects of OSS have been analyzed and discussed . Roughtly speaking, the emergence of open source programs of high quality in monopolistic software markets entails:

- a positive imitative effect that increases proprietary software quality;
- a reduction in proprietary software house market share and profits;
- whether the proprietary software's network of users is small or largely un-skilled (or both), a reduction in proprietary software price;

nevertheless:

- the proprietary software price increases if the adoption of a closed source program ensures large network effects among locked–in users.

Given our results, a welfare analysis of OSS is not strightforward. First of all, we have to decline what welfare means for members of communities. Consumers' surplus is unlikely a good measure of OS users satisfaction as far as ego-gratification concerns, peer-recognition and ethical motives matter in OS communites. Secondly, even whether we accept a surplus-oriented notion of welfare our results shows that both an higher or a lower price for a proprietary software can follow the emergence of an OSS in the market. In particular, higher software quality and lower prices increases consumers surplus at the expences of software house's profits. Whether the former increases more than the latter, an increase in social welfare should be predictable. Nevertheless, it is also possible that, after the emergence of an OSS, large, locked-in networks of un-skilled users of a proprietary software face higher prices for imitative products. This could be harmful for social welfare. Surely, more investigations are needed on welfare consequences of open source. This is particularly true if open and closed source programs share markets characterized by large networks of locked-in users. However, at the current stage of debate on the issue, any conclusion in favour or not to open source from a social welfare perspective is premature.

Finally, our discussion of OSS dominace solutions shows that with low opportunity costs in OSS learing and use, the market may tip to an outcome in which all software offer an open source. Since, for instance, low opportunity costs may be assured by publicy provided and financed education on open source program (see *alumni effect*), the role of the public sector in fostering open source diffusion may be articulate. Governments can adopt an OSS in public departments and bureaus as well as they can directly subsidize free software foudations or associations of OSS developers or foster OSS use within the educational system. Unfortunately, any conclusion about public policies for open source waits for welfare foundations.

References

Bonaccorsi A., Rossi C. (2003) "Why Open Source can Succeed", *Research Policy*, 32(7), pp.1243-58

Casadesus-Masanell R., Ghemanat P. (2003) "Dynamic Mixed Duopoly: a model motivated by Linux and Windows", Strategy Unit WP 04-012, Graduate School of Business Administration, Harvard University

Chiang A.C. (1994) "*Fundamental Methods of Mathematical Economics*", McGraw-Hill, London

Gabsewicz J., Thisse J.F. (1979) "Price Competition, Quality and Income Disparities", *Journal of Economic Theory*, 20(3), pp.340-59

Gaudel A. (2004) "Competition between Open Source and Proprietary Software: The Latex Case Study", unpublished paper, University of Toulose

GNU Project (2000) "*What is Copyleft ?*" downloadable from www-gnu.ai.mit.edu/copyleft/copyleft.html

Hawkins R.E. (2004) "The Economics of Open Source Software for a Competitive Firm", *Netnomics,* 6(2), pp.101-16

Hertel G., Nieder S., Herrmann S. (2003) "Motivation of Software Developers in Open Source Projects: An Internet-based Survey of Contributors to the Linux Kernel", *Research Policy*, 32(7), pp.1159-77

Johnson J.P. (2001) "Economics of Open Source Software", MIT WP, n.233/01

Lakhani K.R., Von Hippel E. (2003) "How Open Source Works: Free User-to-User Assistance", *Research Policy*, 32(7), pp.923-43

Lerner J., Tirole J. (2004) "The Economics of Technology Sharing: Open Source and Beyond", NBER, WP 10956

Lerner J., Tirole J. (2002) "Some Simple Economics of Open Source", *The Journal of Industrial Economics*, L(2), pp.197-234

Lerner J., Tirole J. (2001) "The Open Source Movement: Key Research Questions", *European Economic Review*, 45(2), pp.819-26

Lin L. (2004) "Competition between Proprieraty and Open Source Software with Heterogeneous Users", Boston University WP n.145/04

Mustonen M. (2003) "Copyleft- The Economics of Linux and other Open Source Software", *Information Economics and Policy*, 15(2), pp.99-121

Schiff A. (2002) "The Econonomics of Open Source Software: A Survey of the Early Literature", *Review of Network Economics*, 1(1), pp.66-75

Schmidt K.M., Schnitzer M. (2003) "Public Subsidies for Open Source ? Some Economic Policy Issues of the Software Market", *Harvard Journal of Law and Technology*, 16(2), pp.473-502

Shaked A., Sutton J. (1982) "Relaxing Price Competition Through Product Differentiation", *Review of Economic Studies*, 49(1), pp.3-14

Shy O. (2001) "*The Economics of Network Industries*", MIT Press, Cambridge, MA

Van Wegberg M., Berebds P. (2000) "Competing Communities of Users and Developers of Computer Software: Competition between Open Source Software and Commercial Software", NIBOR WP, NIBOR/RM/00/01

Zadek L.A. (1965) "Fuzzy Sets", *Information and Control*, n.8

Zietlyn D. (2003) "Gift Economies in the Development of Open Source Software; Anthropological Reflections", *Research Policy*, 32(7), pp.1287-91