Open Source Licensing and Software Competition

Vidya Atal* and Kameshwari Shankar†

Abstract

This paper compares effort provision by software developers across different types of open source licenses. We focus on two kind of licenses- restrictive licenses (such as the GNU General Public License or GPL) and non-restrictive licenses (such as the Berkeley Software Distribution or BSD license). The former forces all modifications originating from an open source software to be released under the same open terms. In contrast non-restrictive licenses allow such modifications to be made proprietary by the developer. We develop a model of effort provision from software developers under each kind of license and compare the outcome. Further, we introduce competition from an existing software to study how the existence of competition as well as the type of competition affects developers’ incentives under different licenses. We find several testable hypothesis. First, we show that non-restrictive licenses induce greater effort from developers than restrictive open source licenses with or without competition. Second, competition from an existing software weakly lowers developers’ effort relative to no competition. However the difference in effort between non-restrictive and restrictive licenses is lowered by the presence of competition from a proprietary software. Third, effort under a non-restrictive license may be higher or lower with open source competition than with proprietary competition. In contrast a restrictive license always leads to lower effort with open source competition than with proprietary competition. From a welfare standpoint, we show that relative to the efficient effort level, the non-restrictive always induces too much effort, while a restrictive license may lead to too much or too little effort depending on competition.

Keywords: open source software licensing, price competition

JEL Classification Codes: L17, D4, L11

*Department of Economics and Finance, 427 Partridge Hall, Montclair State University, Montclair NJ, 07043. Email: atalv@mail.montclair.edu

†Department of Economics and Business, City College of New York, NAC 5/139C, New York NY 10031. Email: kshankar@ccny.cuny.edu
1 Introduction

The development of open source software (OSS) has presented many different puzzles to economists over the last decade or so. A number of papers have explored the question of why developers voluntarily give up the right to intellectual property by investing in open source software which requires that all innovations and contributions be kept open and public for anyone to use, distribute or modify. Much of the literature on motivations has tended to focus on the most restrictive version of OSS development where the original software as well as every modification resulting from the original code has to be kept open and free of intellectual property claims. This characterization, however, only fits a specific kind of open source license generally referred to as a restrictive OSS license. The most popular license under this definition is the GNU General Public License (GPL).

While the GPL dominates a large proportion of open source projects (approximately 40 percent of all OSS), it is far from being the only kind of open source software being developed. More generally, there is a wide range of license restrictions that characterize OSS development and developers’ motivations are likely to be affected by the specific license terms of the software in question. Some recent papers have differentiated between restrictive open source license such as GPL and non-restrictive open source licenses that allow modifications to a software originating from an open source code, to be made proprietary by developers. There are a number of open source licenses that have this feature, the most prominent being the Berkeley Software Distribution (BSD) license and the Apache license used in Google’s Android operating system. In this paper, we describe a model of developers’ participation in OSS while allowing the type of license to vary between being restrictive as in the GPL and non-restrictive as in the BSD and Apache licenses.

A number of recent papers have empirically examined the relationship between open source license and software development and there is a general consensus that restrictive licenses are associated with lower levels of participation and performance compared to non-restrictive licenses. Subramaniam et al. (2009) find that restrictive licenses are less likely to generate successful projects. Similarly, Comino et al. (2007) and Lerner and Tirole (2005) find that software distributed under restrictive licenses are less likely to reach a mature and stable release. Fershtman and Gandal (2007) find that output per contributor is much higher when the open source license is less restrictive. While there is considerable empirical research on this topic, there has been no formal theoretical model to capture the mechanism that leads to varying outcomes in different licenses. Thus our first goal in this paper is to provide a theoretical framework for understanding the relationship
between developers’ effort provision and OSS licensing. This allows us to examine the social welfare associated with different licenses. Thus we show that while restrictive licenses may be associated with lower effort provision than non-restrictive licenses, they can sometimes be more efficient. Specifically we show that a non-restrictive license leads to over-provision of developer effort relative to what is efficient. Thus the empirical finding that less restrictive licenses lead to more developer participation does not necessarily provide a case against subsidizing restrictive licenses like the GPL. This is an important result because in recent years many countries have adopted policies that favor restrictive open source licensing and there is an ongoing debate about whether such public policy is misguided. For example Schmidt and Schnitzer (2003) question the need for public policy to encourage open source licenses like the GPL. Our paper argues for caution against dismissing restrictive licensing on the basis of the empirical findings.

Secondly we also look at how software competition affects the incentives for effort provision from developers under different licenses. The empirical research described above ignores the competitive environment faced by OSS. We show that competition affects developer incentives and that this effect depends on the open source license adopted. For example, we find that competition always lowers effort in a software project with a non-restrictive license, whereas it may have no effect on incentives when the license is restrictive. Further, the license adopted by the competing software also influences the difference in effort provision between restrictive and non-restrictive licenses. Thus we find that while competition from a proprietary software lowers the difference in effort, competition from another restrictive software license may enhance or mitigate the effort difference.

A number of papers have looked at how proprietary software competes with open source. However, the focus of this literature has been only on restrictive licenses. We find that open source licensing interacts with competition in significant ways to influence the development of open source software.

We begin by developing a basic model of software development where user-developers independently exert costly effort into creating a software under a given license. The developer who generates the highest value from effort wins the market. The terms of the open source license then determine whether the winning developer can appropriate the user value generated to all user developers through a proprietary price. When the license is restrictive, the developer cannot appropriate this surplus from other user-developers. In this case no matter who generates the highest value,

---

1 For example, the city of Munich in Germany migrated from Microsoft operating system to Linux which is licensed under GPL. In the US, the government provides R&D support for projects released under GPL.
2 Atal and Shankar (2014); Athey and Ellison (2014); Kumar et. al. (2011); Casadesus-Masanell and Ghemawat (2006); Mustonen (2003).
all user-developers enjoy use of the software thus creating the conditions for a classic public good under-provision of effort. On the other hand a non-restrictive license allows the winning developer to appropriate all the user surplus generated in the market. While this incentivizes developers to exert more effort than under a restrictive license, relative to the efficient level, there is a tendency to invest too much effort. The over-investment is driven by the negative externality imposed by every developer on the other developers’ chances of winning the market. Thus our first result conforms to much of the empirical finding that developer effort is higher under a non-restrictive than a restrictive open source license. However, we show that both licenses are inefficient and a non-restrictive license may generate a less optimal outcome than a restrictive one.

We then extend the model to consider competition from an existing software. We consider competition from another open source software and competition from a proprietary software to account for the nature of competition. By accounting for the existence and type of competition we derive a number of testable hypotheses. First, we show that competition from a proprietary software tends to lower the effort difference between new open source projects that adopt a non-restrictive license and those that adopt a restrictive license. Second, we also show that competition of every kind always lowers effort into a new software that adopts a non-restrictive license. On the other hand, effort into a new software with a restrictive license is unchanged when it faces competition from a proprietary software. Further when the new software is developed under a restrictive license, effort provision is always lower in the presence of open source competition compared to competition from a proprietary software. On the other hand the effect of the competitor’s license is ambiguous when the software is developed under a non-restrictive license. Thus not only does the presence of competition itself influence effort provision differentially under the two licenses, even the type of competition that the software faces can have important and asymmetric effects. Finally we find that introducing competition alters the welfare prospects of the two licenses. In particular we find that, while over-provision continues to be a problem for non-restrictive licenses, even the restrictive license may over-provide effort in the presence of proprietary competition from another software. In that case a restrictive license dominates a non-restrictive license in terms of efficiency.

To our knowledge, there are three other papers that also provide a theoretical model of open source licenses. Lerner and Tirole (2005) provide a preliminary model to explain how developer motivations differ across different licenses. However, they do not provide a rigorous framework for explaining the efficiency implications of the license restrictions. Gaudeul (2005) examines the choice between GPL and BSD licenses when developers can hijack the project and sell it for a
positive price. She also shows as we do here that the BSD license allows greater effort provision, but also entails the possibility that the project leader loses some profits to developers. Another paper that explores developer investment in restrictive and non-restrictive licenses is D’Antoni and Rossi (2007) who find that compared to non-restrictive licenses, restrictive licenses encourage greater complementary investments that enhance the value of the open source software. Unlike our paper, none of the three papers mentioned here look at the effect of competition on developers’ choices under different open source licenses.

The paper is organized as follows. In Section 2, we describe the model and assumptions. In Section 3, we describe developers’ effort provision under restrictive and non-restrictive licenses in the absence of competition. Section 4 looks at effort provision under two different licensing assumptions for a competitor software. Finally, Section 5 concludes. All proofs are in the appendix.

2 Model

We consider a model of open source software development with \( N \) user-developers. Each developer \( i \) exerts effort \( e_i \geq 0 \) at a cost of \( ce_i \) where \( c > 0 \) is the marginal cost of effort. The output from developer \( i \)’s effort is \( d_i = e_i^\alpha \varepsilon_i \), where \( 0 < \alpha < \frac{1}{\beta} \) and \( \varepsilon_i \sim U \{0, 2\} \) is a stochastic variable that is independently and identically distributed for each developer. Due to the stochastic component, a developer’s effort may result in a lower value (or zero value) or it may exceed the value of her effort if \( \varepsilon_i > 1 \). The value of the software, let’s call it software \( A \), is \( v_A = \max \{d_i\} \), i.e., the highest value of output generated by the developers determines the software’s quality.

There are two kinds of licenses that a software may have - restrictive and non-restrictive denoted by \( r \) and \( nr \) respectively. Under a restrictive license, the software generated by the highest output of developer’s effort is freely available to all users and developers. Moreover, any modifications to the software must also be distributed under a restrictive license and hence will be available at zero price. Under a non-restrictive license, the winning developer who generates the highest value of effort can make the software proprietary and sell it to other users at a price.

There is an existing software, let’s call it \( B \), that may compete with software \( A \). Software \( B \)’s value is \( v_B \) which is exogenously determined and not influenced by the development process in project \( A \). We consider two different licensing terms for software \( B \) as well - open source and proprietary, denoted by \( R \) and \( NR \). This influences the price at which \( v_B \) is sold. Further, to keep the user-developer population symmetric, we assume that Project \( B \) also comprises of \( N \)
user-developers associated with the project with one of those developers holding the proprietary rights to the software in case B has a proprietary license. Thus the potential number of users in the market is $2N$ with $N$ developers from each project.

The timing of the game is as follows. In stage 1, developers exert effort to develop software A. After effort is provided, the software value, $v_A$, is determined. In stage 2, depending on the licensing terms, the price for software A and B are set by the winning developer in each case. User developers make their consumption choices based on the prices and values generated by the two software. We restrict our attention to symmetric equilibrium in effort provision. The solution concept is Subgame Perfect Nash Equilibrium.

Below, in order to set a benchmark to study the effect of competition and open source licensing, we being by looking at effort provision under each license when there is no competing software in the market. We then introduce competition from software B under different licensing terms to analyze its effect on effort provision and efficiency.

## 3 Effort in the Absence of Competition

In this section, we consider the case where software A does not compete with any other software in the market. In other words, A provides a unique value to its users that cannot be obtained from any other existing software in the market. Hence, for this case, we ignore the value of software B altogether, or alternatively, we set $v_B = 0$. Since the licensing terms determine the winning and losing developers' payoffs after the software is produced, it affects the incentives for effort provision in stage 1.

### 3.1 Restrictive License

Let us start with the case of a restrictive license for Project A. Under a GPL license, a developer's value from effort is $E(v_A)$. Suppose that the other $(N-1)$ developers in Project A exert effort $e_j = e_r$, and $i$ chooses effort $e_i$, then let us calculate the expected value of $v_A$. Define $x = \max\left\{e_i^a, e_r^a, (e_j)_{j \neq i}\right\}$. Then $G(x) = \Pr(d_A < x)$. We can consider two cases for the payoff to developer $i$ given $e_r$.

If $e_i < e_r$, then the p.d.f. is

$$g(x) = \begin{cases} 
  g_1(x) & \text{if } 0 < x < 2e_i^a, \\
  g_2(x) & \text{if } 2e_i^a < x < 2e_r^a,
\end{cases}$$
where

\[ g_1(x) = \frac{Nx^{N-1}}{2e_i^\alpha} \left( \frac{1}{2e_r^\alpha} \right)^{N-1}, \]

\[ g_2(x) = (N-1)x^{N-2} \left( \frac{1}{2e_r^\alpha} \right)^{N-1}. \]

If \( e_i \geq e_r \), then

\[ g(x) = \begin{cases} 
  g_1(x) & \text{if } 0 < x < 2e_r^\alpha \\
  \frac{1}{2e_i^\alpha} & \text{if } 2e_r^\alpha < x < 2e_i^\alpha
\end{cases} \]

Thus the expected payoff to developer \( i \) is:

\[ \pi_i(e_i, e_r) = \begin{cases} 
  2e_i^\alpha \int_0^{2e_r^\alpha} xg_1(x) \, dx + 2e_r^\alpha \int_0^{2e_r^\alpha} xg_2(x) \, dx - ce_i & \text{if } e_i < e_r, \\
  \int_0^{2e_r^\alpha} xg_1(x) \, dx + \frac{2e_r^\alpha}{2e_i^\alpha} \int_0^{2e_r^\alpha} x \, dx & \text{if } e_i \geq e_r.
\end{cases} \]

In a symmetric equilibrium, where all developers choose \( e_i = e_r \), the First Order Condition gives \( e_r \) as

\[ e_r = \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1}{1-\alpha}}. \tag{1} \]

As a benchmark, let us compare this effort level to the first-best effort provision. In the first best level of effort, the social planner will consider the expected value generated to all \( 2N \) users from developer \( i \)'s effort. So the marginal social surplus from a developer’s effort \( e_o \) is

\[ 4N\alpha(e_o)^{-(1-\alpha)} - c. \]

Solving this, we get:

\[ e_o = \left[ \frac{4N\alpha}{(N+1)c} \right]^{\frac{1}{1-\alpha}}. \tag{2} \]

It is evident from (1) and (2) that a restrictive license produces effort provision below the efficient level. This arises from the standard public good under-provision problem as the restrictive license does not allow any developer to appropriate the value of her effort provided to the other developers.

### 3.2 Non-restrictive License

Next let us consider effort provision under a non-restrictive license where the winning developer is allowed to make her innovation proprietary and thus exclude other user-developers’ open access to her software. In the absence of competition, this would make the winning developer a monopolist who sets a price that captures all the user surplus in the market. As a result, given the realized
value of the software $v_A$, if developer $i$ is the winner, she gets $2Nv_A$; but if she does not provide the highest value innovation, her user surplus is zero. Thus the expected payoff to developer $i$ from investing $e_i$ in effort given that the other $(N - 1)$ developers invest $e_{nr}$ is as follows:

$$\pi_i(e_i, e_{nr}) = \begin{cases} 
2N \int_0^{2(e_{nr})^n} e_i^{\alpha} \left( \frac{e_i}{(e_{nr})^n} \right)^{N-1} \frac{de_i}{\lambda} - ce_i & \text{if } e_i < e_{nr}, \\
2N \int_0^{2(e_{nr})^n} e_i^{\alpha} \left( \frac{e_i}{(e_{nr})^n} \right)^{N-1} \frac{de_i}{\lambda} + 2N \int_0^{2(e_{nr})^n} e_i^{\alpha} \frac{de_i}{\lambda} - ce_i & \text{if } e_i \geq e_{nr}.
\end{cases}$$

In a symmetric equilibrium, $\left[ \frac{d\pi_i(e_i, e_{nr})}{de_i} \right]_{e_i=e_{nr}} = 0$, i.e.,

$$e_{nr} = \left[ \frac{4N^2\alpha}{c(N+1)} \right]^{\frac{1}{1-\alpha}}. \quad (3)$$

The following proposition compares effort provision under the restrictive and non-restrictive licenses and also evaluates the social welfare associated with each type of license.

**Proposition 1**  

a) Developer’s effort under a restrictive license is strictly lower than effort under a non-restrictive license, i.e., $e_r < e_{nr}$.

b) Comparing equilibrium effort under the two licenses with the efficient level of effort, a non-restrictive license over-provides effort, while a restrictive license under-provides effort, i.e., $e_r < e_o < e_{nr}$.

Proposition 1 supports the empirical finding that developer participation is lower under restrictive licenses than under non-restrictive licenses. However, the proposition highlights the fact that neither license provides the optimal level of effort relative to cost. Public good characteristics of a restrictive license tend to depress effort below the efficient level, as developers try to free-ride on each other’s effort. At the same time, a non-restrictive license creates a tournament among developers leading to an over-investment by the losing developers. Thus it is unclear whether the non-restrictive license is more optimal than the restrictive license.

### 4 Competition and Effort Provision

In this section, we look at how competition from other software interacts with licensing and affect the effort provision. We consider two cases for the licensing of the competing software - one where the existing product has a restrictive open source and second when it has a non-restrictive or
proprietary. We take the value of the existing software \( (B) \) as exogenously given. Although we do not model the stage 1 development process for the existing software, we can envision a similar development process among \( N \) other developers associated with project \( B \) exerting effort with a winner emerging who then uses the license terms of software \( B \) to make decision regarding its use. We assume that such a development process for \( B \) has been undertaken before development of project \( A \) is initiated and so the developers in \( A \) already know the realized value \( v_B \). In this sense, we take \( v_B \) as exogenous to the development process in \( A \). Since we consider two different licenses for each software, in terms of notation, we identify \( B \)'s license as \( R \) and \( NR \) for restrictive and non-restrictive respectively, while we use lower case for software \( A \)'s license, with \( r \) and \( nr \) respectively. Thus \( e_i^{NR} \) denotes effort in the development of \( A \) when \( A \) has a restrictive license and \( B \) has non-restrictive license. The effort terms for the remaining three situations are defined analogously. Also in order to simplify the exposition we refer to an existing software with a \( NR \) license as proprietary competition and competition from a software with a \( R \) license as open source competition. We can do this without loss of generality since we do not model the development process for software \( B \). Thus even if \( B \) has a non-restrictive license where the first stage of effort provision was kept open, as long as the second stage software is proprietary there is no difference in our analysis.

### 4.1 Competition from an Open Source Software

When software \( B \) is open source it will be available for free to all \( 2N \) developers, i.e., \( p_B = 0 \). If the new software uses a restrictive license, then developers will simply use the higher value software. As a result a developer \( i \)'s payoff in stage 2 is \( \pi_{2i} = \max \{ v_A, v_B \} \) whether she provides the highest value to \( A \) or not.

Given this stage payoff, the expected stage 1 payoff to developer \( i \) from \( e_i \) given that all other developers choose \( e_j = e_i^R \) can be derived as follows.

If \( e_i^R < \left( \frac{v_B}{2} \right)^\frac{1}{\alpha} \), then \( v_A < v_B \) for \( e_i < \left( \frac{v_B}{2} \right)^\frac{1}{\alpha} \). So the expected payoff to developer \( i \) is:

\[
\pi_i (e_i, e_i^R) = \begin{cases} 
  v_B - c e_i & \text{if } e_i < \left( \frac{v_B}{2} \right)^\frac{1}{\alpha}, \\
  \left[ \frac{2(\frac{v_B}{2})^\alpha}{\alpha} \int_0^{g_1(x)} v_B dx + \frac{v_B}{2(\frac{v_B}{2})^\alpha} \int_0^{\frac{v_B}{2}} dx + \int_{v_B}^{2e_i^\alpha} dx \right] - c e_i & \text{if } e_i \geq \left( \frac{v_B}{2} \right)^\frac{1}{\alpha}.
\end{cases}
\]
On the other hand, if \( e^R_i \geq \left( \frac{v_B}{2} \right)^{\alpha/\beta} \), then the expected payoff is:

\[
\pi_i(e_i, e^R) = \begin{cases} \\
\int_0^{v_B} g_1(x) v_B dx + \int_0^{v_B} g_2(x) v_B dx + \int_0^{v^B} x g_2(x) dx \right) - ce_i \text{ if } e_i < \left( \frac{v_B}{2} \right)^{\alpha/\beta} \leq e^R, \\
\int_0^{v_B} g_1(x) v_B dx + \int_0^{v_B} x g_1(x) dx + \int_0^{v^B} x g_2(x) dx \right) - ce_i \text{ if } \left( \frac{v_B}{2} \right)^{\alpha/\beta} \leq e_i < e^R, \\
\int_0^{v_B} g_1(x) v_B dx + \int_0^{v_B} x g_1(x) dx + \int_0^{v^B} \frac{v_B}{2} (e^R)^\alpha x dx \right) - ce_i \text{ if } e_i \geq e^R. \\
\end{cases}
\]

In this case, we can have two potential symmetric equilibria. One where \( e^R_i < \left( \frac{v_B}{2} \right)^{\alpha/\beta} \) for all \( i \) and the other where \( e^R_i \geq \left( \frac{v_B}{2} \right)^{\alpha/\beta} \) for all \( i \). In the former equilibrium, it has to be the case that \( e^R = 0 \) since payoff does not depend on developer’s effort when all developers choose \( e^R_i < \left( \frac{v_B}{2} \right)^{\alpha/\beta} \). This will be an equilibrium when cost of effort is very high. In order to rule out the equilibrium where the software is never developed under a restrictive license, we assume that costs are low enough to allow the other equilibrium with \( e^R_i \geq \left( \frac{v_B}{2} \right)^{\alpha/\beta} \) to exist.\(^3\)

The symmetric equilibrium, where \( e^R_i \geq \left( \frac{v_B}{2} \right)^{\alpha/\beta} \), solves the First Order Condition \( \left[ \frac{d\pi_i(e_i, e^R_i)}{de_i} \right]_{e_i = e^R} = 0 \), or

\[
2 \frac{\alpha}{N+1} (e^R)^{-(1-\alpha)} \left[ 1 - \left( \frac{v_B}{2(e^R)^{\alpha}} \right)^{N+1} \right] = c = 0. 
\]

**Lemma 1** Suppose both software A and software B adopt a restrictive license, then there exists \( c_1 > 0 \) such that if \( c < c_1 \), then there is a unique symmetric equilibrium where all developers exert a positive level of effort in stage 1 and this equilibrium effort solves (4).

In looking at the efficient effort level, observe that the ex post efficient value derived by each developer is the same as when a restrictive license is used in both software, i.e., it is max \( \{v_A, v_B\} \). However, once again, the public good under-provision takes over in stage 1 so the social value of developer’s effort is \( 2N \) times each developer’s value. Thus the efficient effort level for software A in the presence of a substitute with value \( v_B \) is \( e^co \) where

\[
\frac{4N\alpha}{N+1} (e^co)^{-(1-\alpha)} \left[ 1 - \left( \frac{v_B}{2(e^co)^{\alpha}} \right)^{N+1} \right] = c = 0. 
\]

Clearly, comparing (4) and (5), we see that \( e^co > e^R_i \). Similarly, comparing \( e^R_i \) and \( e^co \) from (4) and (1), we see that competition from an existing restrictively licensed software reduces effort from developers. These results are stated in the proposition below.

\(^3\)The details of the restriction on \( c \) is provided in the appendix in the proof of Lemma 1.
Proposition 2 Suppose an existing software (B) is open source, then the following is true about the effort provision in a new software (A) with a restrictive license.

a) Effort to develop A continues to be less than the efficient level, i.e., \( e^R_1 < e^\alpha \).

b) Effort is lower than the case where A did not face any competition, i.e., \( e^R_1 < e_f \).

Proposition 2 shows that the existence of an alternative software lowers the incentive for effort into developing a new software. This is because, if the value of effort does not exceed a threshold, the effort from the developers is wasted. As a result, the marginal return to effort is lowered in the presence of competition.

Next suppose project A has a non-restrictive license, then stage 2 profits to the winning developer is limited by the availability of the free alternative software B in the market. Specifically, if \( v_A \leq v_B \), then everyone uses software B so that the winning developer in A does not make any profits and she gets a surplus of \( v_B \) as a user. On the other hand, if \( v_A > v_B \), then the winning developer in A can charge a price of \( p_A = (v_A - v_B) \) and her total payoff is the sum of her surplus as a user of software A and her profits from selling A to the remaining \( 2N - 1 \) developers, i.e., \( v_A + (2N - 1) (v_A - v_B) \). If developer \( i \) does not produce the highest value but some other developer within the project does, so that \( v_A > v_B \), then \( i \) will pay a price of \( (v_A - v_B) \) to the winning developer so that her surplus as a user in stage 2 is \( v_B \).

Proceeding to stage 1 effort provision, consider the payoff to developer \( i \) from \( e_i \) given that all other developers choose \( e^R_{nr} \). Now in order to win the market for the software in stage 1, the value generated through developer \( i \)'s effort must beat every other developer’s value and \( v_B \), i.e.,

\[
e_i^\alpha > (e^R_{nr})^\alpha \varepsilon_j \quad \text{for all } j \neq i \quad \text{and} \quad e_i^\alpha \varepsilon_i > v_B.
\]

If \( e_i < \left( \frac{v_B}{2} \right)^\frac{1}{\alpha} \) then \( i \) will never win the market. For \( e_i \geq \left( \frac{v_B}{2} \right)^\frac{1}{\alpha} \), define the stage 2 payoffs when \( i \) wins the market given \( e_i \) as:

\[
\tilde{\pi}_2^A \left( e_i | \varepsilon_i \right) > \left[ \frac{e_i^\alpha \varepsilon_i + (2N - 1) (e_i^\alpha \varepsilon_i - v_B)}{2} \varepsilon_i \right]^{N-1}.
\]

If \( e^R_{nr} < \left( \frac{v_B}{2} \right)^\frac{1}{\alpha} \), then as long as \( e_i > \left( \frac{v_B}{2} \right)^\frac{1}{\alpha} \), \( i \) will win the market in stage 2 so her stage 1 expected payoff is:

\[
\pi_i \left( e_i, e^R_{nr} \right) = \left\{ \begin{array}{ll}
\frac{v_B}{2} d_{e_i} & \text{if } e_i^\alpha < \frac{v_B}{2}, \\
\int_0^{v_B} \frac{d_{e_i}}{2} + \left[ \int_{e_i^\alpha}^{v_B} \frac{d_{e_i}}{2} \right]^{N-1} & \text{if } e_i^\alpha \geq \frac{v_B}{2}.
\end{array} \right.
\]

11
On the other hand, if $e_{nr}^R \geq (\frac{v_B}{2})^{\frac{1}{\alpha}}$, the expected payoff is:

$$\pi_i (e_i, e_{nr}^R) = \begin{cases} 
    v_B - ce_i & \text{if } e_i < (\frac{v_B}{2})^{\frac{1}{\alpha}}, \\
    \frac{v_B}{\alpha} \int E (v_A) \frac{d\xi}{2} + \frac{2}{\alpha} \pi^A (e_i | \xi_i \leq 2(e_{nr}^R)^{\alpha}) \frac{d\xi}{2} - ce_i & \text{if } (\frac{v_B}{2})^{\frac{1}{\alpha}} \leq e_i < e_{nr}^R, \\
    \frac{2(e_{nr}^R)^{\alpha}}{\alpha} \int E (v_A) \frac{d\xi}{2} + \int \frac{2}{\alpha} \pi^A (e_i | \xi_i \leq 2(e_{nr}^R)^{\alpha}) \frac{d\xi}{2} & \text{if } e_i \geq e_{nr}^R,
\end{cases}$$

where $E (v_A)$ is the expected value of $\max \{ e_i^O, e_r (\varepsilon_j)_{j \neq i} \}$ as derived in Section 3.1.

Again, as with the restrictive license, it is possible for effort to be zero in equilibrium when costs are high enough. So we restrict $c$ to be low enough so that development always occurs at the symmetric equilibrium.

Under this restriction, the symmetric equilibrium with positive effort, $e_{nr}^R$ solves the First Order Condition

$$4\alpha [e_{nr}^R]^{-(1-\alpha)} \left\{ \frac{N^2}{N+1} - \frac{1}{N+1} \left( \frac{v_B}{2(e_{nr}^R)^{\alpha}} \right)^{N+1} - (N-1) \frac{v_B}{2(e_{nr}^R)^{\alpha}} \right\} - c = 0. \quad (6)$$

The lemma below states the conditions for a symmetric interior equilibrium in effort.

**Lemma 2** Suppose software A adopts a non-restrictive license and software B has a restrictive license, then there exists $c_2 > 0$ such that if $c < c_2$, then there is a unique symmetric equilibrium where all developers in A exert positive effort in stage 1 and it solves (6).

Comparing (6) and (4), as before, we continue to find that effort is higher under a non-restrictive license than under a restrictive license. Similarly, from (6) and (5), over-provision of effort continues to make the non-restrictive license inefficient. Further, we also see that competition depresses effort due to the possibility that the ex post user value is determined by the alternative software which is independent of effort provision. These results are summarized in the proposition below.

**Proposition 3** Suppose an existing software (B) is open source, then the following is true about the effort provision in a new software (A) with a non-restrictive license.

a) Effort is lower than in the absence of competition, i.e., $e_{nr} > e_{nr}^R$.

b) Effort under the new software is greater when it adopts a non-restrictive license than under a restrictive license, i.e., $e_{nr}^R > e_{nr}$.

c) Effort is higher than the efficient level of effort, i.e., $e_{nr}^R > e^O$. 

12
From Propositions 2 and 3, we find that competition from a restrictive license unambiguously lowers effort in a new software and this is true independent of the license adopted by the new software. At the same time, the qualitative differences in effort levels between a restrictive license, a non-restrictive license and the efficient outcome are the same as in the case where there was no competition. Thus, as in the case of no competition, it is not clear which open source license is superior in terms of efficiency. Even though the restrictive license is associated with less effort than the non-restrictive license, it is possible for it to be more efficient.

4.2 Competition from Proprietary Software

Let us consider the case where the existing software is sold as a proprietary product at a positive price. If the new project adopts a restrictive license then by definition the price of the new software is $p_A = 0$. Since developers can always consume the new software for free and earn a consumer surplus of $v_A$, the proprietary seller of $B$ must set a price of $p_B = \max\{v_B - v_A, 0\}$.

Thus with a restrictive license, the developer’s user value is always $v_A$ which is the same as the case when there was no competition. This means that effort provision under a restrictive license is not affected by the presence of competition from a proprietary software. Also since we do not model the development process for software $B$, there is no difference theoretically between a non-restrictive license and a proprietary license for $B$. Hence, when it has a non-restrictive license, we refer to competition from $B$ as a proprietary competition without loss of generality.

Proposition 4 Suppose the competing software is proprietary then the following holds:

a) Effort provision in a new software with a restrictive open source license is the same as the effort under no competition, i.e., $e^NR_i = e_r$.

b) The restrictive license in a new software under-provides effort relative to the efficient level if and only if the marginal cost of effort is low enough, i.e. $e^NR_i \leq e^o$ if and only if $c \leq \frac{2^\alpha}{(N+1)} \left( \frac{2}{v_B} \right)^{1-\alpha} \left[ \frac{2N-1}{2N} \right]^{1-\alpha} \left( \frac{N+1}{\alpha} \right)^{-\alpha}$.

Proposition 4 highlights an important feature of restrictive licenses which is that its development is not affected by proprietary competition. As we show below, this is not true for non-restrictive licenses where competition always has a tampering affect on developers’ effort.

\[4\] We make the following indifference assumptions for users. If a user is indifferent between buying and not buying she always buys the product. If she is indifferent between the lower quality product and the higher quality software, she chooses the latter.
When the new software is developed under a non-restrictive license there is price competition between two proprietary sellers in stage 2 after the value of \( v_A \) is realized. The lemma below characterizes the price equilibrium that emerges from this stage 2 competition between \( A \) and \( B \). We can consider three cases for developer \( i \) in stage 2. First, if she is the winning developer and \( v_A > v_B \), then she can charge a price of \( p_A = (v_A - v_B) \) to the remaining \( (2N-1) \) users by making her innovation proprietary. In that case her total payoff is \( v_A + (2N-1)(v_A - v_B) \). Second, if developer \( i \) does not win the market but someone else from project \( A \) wins the market, then she will use software \( A \) at a price of \( (v_A - v_B) \) so that her user surplus is \( v_B \). Finally, if no one from project \( A \) wins the market then the superior software is \( B \) which is sold at a price of \( p_B = (v_B - v_A) \) and \( i \)'s user payoff is \( v_A \).

As before, consider the expected payoff from effort to developer \( i \), given that all the other developers \( j \neq i \) choose \( e_j = e_{NR}^j \). As in the case of competition from a restrictive license, \( i \) will never win the market if \( e_i < \left( \frac{v_B}{2} \right)^{\frac{1}{a}} \). If \( e_i \geq \left( \frac{v_B}{2} \right)^{\frac{1}{a}} \), then we define her expected payoff from winning the market given \( e_i > \frac{v_B}{e_i} \) as \( \pi_1^A \left( e_i \right) \) and \( \pi_2^A \left( e_i \right) \) as before.

If \( e_{NR}^i \left( \frac{v_B}{2} \right)^{\frac{1}{a}} \), then \( i \) will win the market as long as \( e_i > \frac{v_B}{e_i} \). If she loses the market, then she will buy from \( B \) and receive a payoff of \( v_A \). Thus her expected payoff is:

\[
\pi_i \left( e_i, e_{NR}^i \right) = \begin{cases} 
E \left( v_A \right) - ce_i & \text{if } e_i < \left( \frac{v_B}{2} \right)^{\frac{1}{a}} , \\
\int_0^{\frac{v_B}{e_i}} E \left( v_A \right) \frac{dx}{2} + \int_{\frac{v_B}{e_i}}^{\infty} \pi_2^A \left( e_i | e_i > \frac{2(e_{NR}^i)^\alpha}{e_i^\alpha} \right) \frac{dx}{2} - ce_i & \text{if } e_i \geq \left( \frac{v_B}{2} \right)^{\frac{1}{a}} .
\end{cases}
\]

If \( e_{NR}^i \geq \left( \frac{v_B}{2} \right)^{\frac{1}{a}} \), then \( i \) will never win the market if \( e_i < \left( \frac{v_B}{2} \right)^{\frac{1}{a}} \), however her payoff in stage 2 depends on whether \( v_A > v_B \) or \( v_A \leq v_B \). If \( \left( e_{NR}^j \right)^\alpha e_j < v_B \) for all \( j \neq i \), so that \( v_A < v_B \), then \( \pi_2^i \left( e_i, e_{NR}^i \right) = \max_{k \in \{1,2,\ldots,N\}} \{ e_k^\alpha \} \). But if \( \left( e_{NR}^j \right)^\alpha e_j \geq v_B \) for some \( j \), then \( \pi_2^i \left( e_i, e_{NR}^i \right) = v_B \).

For \( e_j < \frac{v_B}{(e_{NR}^i)^\alpha} \), let us define \( x = \max_{k \in \{1,2,\ldots,N\}} \{ e_k^\alpha \} \). Then the p.d.f. of \( x \) is as follows:

\[
f(x) = \begin{cases} 
f_1(x) & \text{if } x < 2e_i^\alpha , \\
f_2(x) & \text{if } 2e_i^\alpha < x < v_B ,
\end{cases}
\]

where

\[
f_1(x) = \frac{N}{2e_i^\alpha} \left( \frac{x}{v_B} \right)^{N-1},
\]

and

\[
f_2(x) = (N - 1)x^{N-2} \left( \frac{1}{v_B} \right)^{N-1}.
\]
If $e_i^* \geq \frac{v_B}{2}$, then the p.d.f. for $x \in [0, v_B]$ given that $\varepsilon_i < \frac{v_B}{e_i^*}$ and $\varepsilon_j < \frac{v_B}{(e_b^*)^\alpha}$ for all $j \neq i$ is:

$$f_3(x) = \frac{N}{v_B} \left( \frac{x}{v_B} \right)^{N-1},$$

with

$$E(x) = \frac{Nv_B}{(N+1)}.$$ 

Thus the expected payoff to developer $i$, $\pi_i(e_i, e_{nr}^{NR})$, is:

$$\left[ 1 - \left( \frac{v_B}{2(e_b^*)^\alpha} \right)^N \right] v_B + \left\{ \begin{array}{ll}
\left( \frac{v_B}{2(e_b^*)^\alpha} \right)^N \left[ \int_0^{v_B} x f_1(x) \, dx + \int_0^{v_B} x f_2(x) \, dx \right] - ce_i \text{ if } e_i < \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}}, \\
\int_0^{v_B} \left( \frac{v_B}{2(e_b^*)^\alpha} \right)^N \left( \frac{v_B}{N+1} \right)^{N-1} \frac{Nv_B}{v_B} \frac{d\varepsilon_i}{(N+1)^2} + \frac{2}{\pi^2} \int_0^{v_B} \left( e_i \left| \varepsilon_i \leq \frac{2(e_b^*)^\alpha}{e_i} \right. \right)^{N+1} - ce_i \text{ if } e_i \geq \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}}. 
\end{array} \right.$$ 

Once again we have two possibilities for the equilibria. If $e_{nr}^{NR} < \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}}$, then it is evident that the payoff to the developer is identical to the payoff in the restrictive license case. Hence if a symmetric equilibrium exists in this range, then it must be the case that $e_{nr}^{NR} = e_r^{NR}$, i.e., effort provision is the same as under a restrictive license. This equilibrium will exist if $c$ is high.

On the other hand, if in equilibrium $e_{nr}^{NR} \geq \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}}$, then it solves the First Order Condition where

$$\left[ \frac{d\pi_i(e_i, e_{nr}^{NR})}{d\varepsilon_i} \right]_{e_{nr}^{NR}} = 0 \text{ or,}$$

$$2\alpha \left( e_{nr}^{NR} \right)^{(1-\alpha)} \left\{ \frac{(2N-1)N+1}{N+1} - \frac{(2N-1)}{(N+1)N} \left( \frac{v_B}{2(e_{nr}^{NR})^\alpha} \right)^N \right\} - \frac{(2N-1)(N-1)}{N} \frac{v_B}{2(e_{nr}^{NR})^\alpha} = c.$$ 

(7)

The following proposition describes the equilibrium effort provision when both the existing and the new software have a non-restrictive license.

**Proposition 5** Suppose the existing software is proprietary, then for $c < \frac{2\alpha}{(N+1)} \left( \frac{v_B}{2} \right)^{1-\alpha}$, effort to develop a new software under a non-restrictive license is strictly higher than the effort under a restrictive license, i.e., $e_{nr}^{NR} > e_r^{R}$. However, if $c \geq \frac{2\alpha}{(N+1)} \left( \frac{v_B}{2} \right)^{1-\alpha}$, then effort provision in a new software does not depend on the license adopted by the new software, i.e., $e_{nr}^{NR} = e_r^{NR} = e_r$.

This result provides an interesting hypothesis that can be empirically tested. In the presence of competition from proprietary software, open source license is less likely to affect effort provision from developers. Below we compare $e_{nr}^{NR}$ to $e_r^{NR}$.
**Proposition 6** Suppose the new software faces proprietary competition and is developed under a non-restrictive license, then the following is true about effort provision.

a) Effort is lower relative to the case where there was no competition, i.e., $e_{nr}^{NR} < e_{nr}$.

b) There exists $c_3 > 0$ such that effort is higher relative to competition from a restrictive license if and only if $c > c_3$, i.e., $e_{nr}^{NR} > e_{nr}^R$ if and only if $c > c_3$.

c) There is over-provision of effort relative to the efficient level, i.e., $e_{nr}^{NR} > e^c$.

Proposition 6 provides a contrast to the results from Propositions 2 and 4. Compared to proprietary competition, competition from a restrictive license always lowers effort into developing a new software under a restrictive license as the free alternative in the latter case exacerbates the tendency to free-ride. On the other hand, proprietary competition can depress or intensify effort into developing software under a non-restrictive license depending on the cost of effort.

Further, we also see that although a nonrestrictive license continues to incentivize greater effort from developers than a restrictive license, the difference in effort provision between the two licenses is lower with proprietary competition than when the software faced no competition. This is because effort under the restrictive license is the same while effort under a non-restrictive license decreases, thereby reducing the overall difference. For costs low enough, we see that the difference in effort between the two licenses is smaller with proprietary competition than with restrictive license competition as well. These results are stated in the proposition below.

**Proposition 7** Suppose the competing software, B, is proprietary, then the following is true about the difference in effort provision between a non-restrictive and restrictive license in the development of software A.

a) The effort difference is lower than when A faced no competition, i.e., $e_{nr}^{NR} - e_{nr}^{NR} < e_{nr} - e_r$.

b) If $c < c_3$, then the effort difference is lower than if B is open source, i.e., $e_{nr}^{NR} - e_{nr}^{NR} < e_{nr}^R - e_r$.

Finally, we can also provide some conclusive welfare implications of the software license when competition comes from a non-restrictive license. While in absolute terms effort provision continues to be inefficient under both licenses, we can compare the relative efficiency between the two licenses with proprietary competition. We see that with proprietary competition, when effort cost is high, both types of licenses over-provide effort, although this over-provision is lower under a restrictive license. This makes the restrictive license superior from a second-best efficiency perspective.

**Proposition 8** If $c > \frac{2\alpha}{(N+1)} \left(\frac{2}{\theta}\right)^{\frac{1-\alpha}{\alpha}} \left[\frac{2N-1}{2N}\right]^{\frac{1-\alpha}{\alpha}}$, and software A faces proprietary competition from B, then a restrictive license for A is more efficient than a non-restrictive license.
Proposition 8 argues that even though a non-restrictive license may induce greater participation from developers than a restrictive license, it may not be efficient. In particular, non-restrictive licenses suffer from over-participation relative to costs and hence a restrictive license may be more optimal. This is especially true when the OSS faces proprietary competition. Thus the result argues for caution against advocating for non-restrictive licenses on the basis of the empirical findings regarding developer’s participation.

5 Conclusion

We compared effort provision by software developers across two types of open source licenses - restrictive licenses that force all modifications originating from an open source software to be released under the same open terms and non-restrictive licenses that allow such modifications to be made proprietary by the developer. We provided a model of developer effort provision under each kind of license. We found that effort under a non-restrictive license is greater than under a restrictive license thus supporting the overwhelming empirical finding that restrictive licenses adversely affect developer participation. Further, we studied the impact of competition under different licenses by looking at effort provision in the presence of proprietary and open source competition. We found that the existence of competition and the nature of competition affects effort provision under the two licenses and moreover these effects are sometimes asymmetric. We also derived policy implications for open source licensing. We showed that a non-restrictive license always leads to over-provision of effort relative to the efficient level of effort. On the other hand a restrictive license may lead to higher or lower effort than the efficient level. We find that the efficient license choice among the two licenses is ambiguous. An important extension of our paper is to consider the issue of license choice within our framework. Currently we assume that software license is exogenously given and hence our model only describes the causal effect of license on effort provision. However, project leaders often try to pick the best license for their software project. Characterizing license choice endogenously in our model is likely to provide useful insights about when and where certain open source licenses are more likely to exist.

Appendix

Proof of Proposition 1. a) Comparing (1) and (3), we see that \( \left[ \frac{4N^2\alpha}{c(N+1)} \right]^{\frac{1}{1-\alpha}} < \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1}{1-\alpha}} \) and hence \( e_{nr} < e_r \).

b) Comparing (1) and (2), \( \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1}{1-\alpha}} < \left[ \frac{4N\alpha}{(N+1)c} \right]^{\frac{1}{1-\alpha}} \) so \( e_r < e^\alpha \). From (3) and (2), \( \left[ \frac{4N^2\alpha}{c(N+1)} \right]^{\frac{1}{1-\alpha}} > \)
First let us consider the equilibrium where $e^R_r = 0$. Looking at the payoff function where $e^R_r < \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}}$, $\frac{d^2}{de^R_i} \pi_i (e_i, 0) > 0$ if and only if $e_i < \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}} \frac{1}{(1+\alpha) \left( \frac{1}{1-\alpha} \right)}$. Thus $\frac{d}{de^R_i} \pi_i (e_i, 0)$ is first $-\kappa$, then increases until $e_i = \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}} \frac{1}{(1+\alpha) \left( \frac{1}{1-\alpha} \right)}$ where it reaches the maximum and then starts to fall. In order for $e^R_r = 0$ to be an equilibrium, $Max \pi_i (e_i, 0) < v_B$. This will always be true if

$$\max \left[ \frac{d}{de^R_i} \pi_i (e_i, e^R_r) \right] = \max \left[ \frac{d}{de^R_i} \pi_i (e_i, e^R_r) \right] = \frac{2 \alpha^2}{1+\alpha} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{1+\alpha} \right)^{\frac{1}{2\alpha}} < 0, \text{ or } c > \frac{2 \alpha^2}{1+\alpha} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{1+\alpha} \right)^{\frac{1}{2\alpha}}.$$

If $c < \frac{2 \alpha^2}{1+\alpha} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{1+\alpha} \right)^{\frac{1}{2\alpha}}$, then $\pi_i (e_i, 0)$ reaches a maximum at $\hat{e}$ where $\hat{c}$ solves $\frac{d\pi_i (e^R_i, 0)}{de^R_i} = 0$. Then $Max \pi_i (e_i, 0) < v_B$ if and only if $\pi_i (\hat{e}, 0) < v_B$. This gives the condition $c > \frac{4 \alpha^2}{(1+\alpha)^2} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{1+\alpha} \right)^{\frac{1}{\alpha}}.$

Thus if $c < \bar{c}_1$ where $\bar{c}_1 = \min \left\{ \frac{2 \alpha^2}{1+\alpha} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{1+\alpha} \right)^{\frac{1}{2\alpha}}, \frac{4 \alpha^2}{(1+\alpha)^2} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{1+\alpha} \right)^{\frac{1}{\alpha}} \right\}$ then $e^R_r$ cannot be an equilibrium.

Now let us look at the other equilibrium where $e^R_r \geq \left( \frac{v_B}{2} \right)^{\frac{1}{\alpha}}$. Looking at the payoff function in this range, at $e_i = e^R_r$, $\frac{d^2}{de^R_i} \pi_i (e_i, e^R_r) < 0$ for $e^R_r$ high enough. Thus $\frac{d}{de^R_i} \pi_i (e^R_r, e^R_r)$ first decreases then increases, reaches a maxima and then decreases. Hence for a symmetric equilibrium with positive effort to exist, $Max \frac{d}{de^R_i} \pi_i (e^R_r, e^R_r) > 0$ and $Max \pi_i (e^R_r, e^R_r) > v_B$. Solving these two conditions gives us $c < \bar{c}_1$ where

$$\bar{c}_1 = \min \left\{ \frac{2 \alpha^2}{1+\alpha} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( 1 - N \alpha \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{2 \rho_1 v}{v_B} \right)^{\frac{1-\alpha}{\alpha} \left( 1 - N \alpha \right)^{\frac{1-\alpha}{\alpha}}}, N \alpha \right\},$$

and $\rho_1$ solves

$$(1 + \alpha) \rho_1^{N+1} + N - \alpha - (N+1) \rho_1 = 0.$$

Define $c_1 = \min \{ \bar{c}_1, \bar{c}_1 \}$. 

Proof of Proposition 2. a) Substituting $e^{co}$ in the FOC for $e^R_r$ given by (4), we get

$$FOC_{e^R_r} (e^{co}) = -\frac{2 (2N - 1) \alpha (e^{co})^{-(1-\alpha)}}{N+1} \left[ 1 - \left( \frac{v_B}{2 (e^{co})^\alpha} \right)^{N+1} \right] < 0.$$

Hence $e^{co} > e^R_r$.

b) Substituting $e_r$ from (1) into FOC for $e^R_r$ given by (4) we get

$$FOC_{e^R_r} (e_r) = -c \left( \frac{v_B}{2 \left( \frac{2 \alpha}{N+1} \right)^{\frac{1-\alpha}{\alpha}}} \right)^{N+1} < 0.$$

Hence $e_r > e^R_r$. 

Proof of Lemma 2. The proof is analogous to the proof of Lemma 2. The conditions under which \( e_{nr}^R = 0 \) cannot be an equilibrium are \( \max \left[ \frac{dr(x,0)}{dx} \right] > 0 \) or \( \max \pi_i (e_i, 0) > v_B \). This gives us the condition \( c < \tilde{c}_2 \), where

\[
\tilde{c}_2 = \min \left\{ 4N \left[ \frac{\alpha^2}{1+\alpha} \left[ \frac{1}{1+\alpha} \sum \right] \frac{1}{2} \left( \frac{2}{v_B} \right)^{\frac{\alpha}{1+\alpha}} \left( \frac{8N \alpha^2}{(1+\alpha)^2} \right) \left( \frac{2}{v_B} \right)^{\frac{\alpha}{1+\alpha}} \right] \right\}
\]

The conditions under which \( e_{nr}^R > \left( \frac{v_B}{\alpha} \right)^{\frac{1}{\alpha}} \) is an equilibrium are \( \max \left[ \frac{dr(x,e_{nr}^R)}{dx} \right] > 0 \) and \( \max \pi_i (e_{nr}^R, e_{nr}^R) > v_B \) which gives us \( c < \tilde{c}_2 \) where

\[
\tilde{c}_2 = \min \left\{ 4N \left[ \frac{\alpha^2}{1+\alpha} \left( \frac{2 \rho_1}{v_B} \right)^{\frac{1}{\alpha}} \left[ N - (N - 1) \rho_2 \right] \right], 4\alpha e_i^{-1+\alpha} \left( \frac{2 \rho_2}{v_B} \right)^{\frac{1}{\alpha}} \left[ \frac{N^2}{N+1} - \frac{1}{N+1} \rho_3^{N+1} - (N - 1) \rho_3 \right] \right\}
\]

\( \rho_2 \) solves

\[-N^2 (1 - N\alpha) + (1 + \alpha) \rho_2^{N+1} + (1 - N\alpha + \alpha) (N^2 - 1) \rho_2 = 0,
\]

and \( \rho_3 \) solves

\[N (1 - N\alpha) + (1 + \alpha) \rho_2^{N+1} - (N + 1) (1 - N\alpha + \alpha) \rho_2 = 0.
\]

Define \( c_2 = \min \{ \tilde{c}_2, \tilde{c}_2 \} \).

Proof of Proposition 3. a) Substituting \( e_{nr} \) in the FOC \( e_{nr}^R \) shown in (6):

\[
FOC_{e_{nr}^R}(e_{nr}) = -c \left[ \frac{1}{N^2} \left( \frac{v_B}{2 \left[ \frac{4N^2 \alpha e_{nr}^{N+1}}{c(N+1)} \right]^{\frac{1}{\alpha}} \right) + \frac{N^2 - 1}{N^2} \right] < 0.
\]

Hence \( e_{nr} > e_{nr}^R \).

b) Substituting for \( e_r \) from (4) in FOC \( e_{nr}^R \), we get,

\[
FOC_{e_{nr}^R}(e_r) = 2a \left[ \frac{(e_r^{(1-\alpha)} - (1-\alpha))}{N+1} \left( \frac{2}{v_B} - \left( \frac{v_B}{2 (e_{nr}^{(1-\alpha)})} \right)^{N+1} \right) - 2 (N^2 - 1) \frac{v_B}{2 (e_{nr}^{(1-\alpha)})} \right] > 0.
\]

Hence \( e_r^R < e_{nr} \).

c) Substituting for \( e^{co} \) from (5) in FOC \( e_{nr}^R \), we get,

\[
FOC_{e_{nr}^R}(e^{co}) = 4 \left[ \frac{(N+1) \alpha (e^{co})^{-(1-\alpha)}}{N+1} \right] \left[ N - (N+1) \frac{v_B}{2 (e_{nr}^{(1-\alpha)})} + \left( \frac{v_B}{2 (e^{co})^{N+1}} \right) \right] > 0.
\]

Hence \( e^{co} < e_{nr} \).

Proof of Proposition 4. a) Since the stage 2 payoff to the developer is the same with or without competition from a non-restrictive licensed software, the stage 1 effort remains the same.
b) Substituting \( e_{r}^{NR} = e_{r} = \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1-\alpha}{\alpha}} \) into the FOC given in (5) we get

\[
FOC_{e^{e^{CO}}} (e_{r}) = c \left[ (2N - 1) - 2N \left( \frac{v_{B}}{2 \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1-\alpha}{\alpha}}} \right)^{N+1} \right].
\]

Hence, \( FOC_{e^{e^{CO}}} (e_{r}) > 0 \) if and only if \( c < \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \left[ \frac{2N-1}{2N} \right]^{\frac{1-\alpha}{\alpha}}. \]

**Proof of Proposition 5.** \( e_{nr}^{NR} = e_{r} = \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1-\alpha}{\alpha}} \) will be an equilibrium if and only if \( \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1-\alpha}{\alpha}} < \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \) or \( c > \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \). Consider the \( FOC_{e_{nr}^{NR}} \) shown in (7). In order for the equilibrium to exist \( FOC_{e_{nr}^{NR}} (e_{r}) > 0 \). This gives the condition \( c < \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}}. \)

**Proof of Proposition 6.** a) When \( c > \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \), and \( e_{nr}^{NR} = e_{r} \), it is clear that \( e_{r} < e_{nr} \).

For the case where \( c < \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \), let us substitute \( e_{nr} \) into the \( FOC_{e_{nr}^{NR}} \) given by (7).

\[
FOC_{e_{nr}^{NR}} (e_{nr}) = -\frac{c}{2N^{2}} \left[ (N - 1) + \frac{(2N - 1)}{N} \left( \frac{v_{B}}{\left[ \frac{4N^{2}\alpha}{c(N+1)} \right]^{\frac{\alpha}{1-\alpha}}} \right)^{N+1} + \frac{(2N - 1)(N-1)}{N(N+1)} \left( \frac{v_{B}}{\left[ \frac{4N^{2}\alpha}{c(N+1)} \right]^{\frac{\alpha}{1-\alpha}}} \right) \right].
\]

Hence \( e_{nr} > e_{nr}^{NR} \).

b) Let us take the \( FOC_{e_{nr}^{R}} \) from (6). For \( c > \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \), \( e_{nr}^{NR} = \left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{1-\alpha}{\alpha}} \) and \( FOC_{e_{nr}^{R}} (e_{nr}^{NR}) \) is

\[
FOC_{e_{nr}^{R}} (e_{nr}^{NR}) = c \left[ (2N^{2} - 1) - 2 \left( \frac{v_{B}}{\left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{\alpha}{1-\alpha}}} \right)^{N+1} - 2 \left( N^{2} - 1 \right) \left( \frac{v_{B}}{\left[ \frac{2\alpha}{(N+1)c} \right]^{\frac{\alpha}{1-\alpha}}} \right) \right] < 0.
\]

Hence \( e_{nr}^{NR} > e_{nr}^{R} \). For \( c < \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \), we substitute \( e_{nr}^{NR} \) from (7) into \( FOC_{e_{nr}^{R}} \).

\[
FOC_{e_{nr}^{R}} (e_{nr}^{NR}) = 2\alpha \left( e_{nr}^{NR} \right)^{(1-\alpha)} \left[ \frac{N - 1}{N+1} - \frac{1}{N} \left( \frac{v_{B}}{2(e_{nr}^{NR})^{\alpha}} \right)^{N+1} - \frac{(N-1)}{N} \frac{v_{B}}{2(e_{nr}^{NR})^{\alpha}} \right].
\]

The above is negative at \( c = \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \), however it is positive for \( c \) low enough. Define \( c_{3} < \frac{2\alpha}{(N+1)c} \left( \frac{2}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \) as

\[
c_{3} = 2\alpha \left( \frac{2\rho_{4}}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \left\{ \frac{(2N-1)N+1}{N+1} - \frac{1}{(N+1)N} \rho_{4}^{N+1} \right\} - \frac{(2N-1)(N-1)}{N} \rho_{4}^{N+1} \]

where \( \rho_{4} \) solves

\[
2\alpha \left( \frac{2\rho_{4}}{v_{B}} \right)^{\frac{1-\alpha}{\alpha}} \left[ \frac{N - 1}{N+1} - \frac{1}{(N+1)N} \rho_{4}^{N+1} - \frac{(N-1)}{N} \rho_{4}^{N+1} \right] = 0.
\]
c) Finally substituting $\epsilon^{co}$ in the $FOC_{enr}$ equation we get

$$FOC_{enr}^{NR} (\epsilon^{co}) = \frac{2}{N+1} \alpha (\epsilon^{co})^{-(1-\alpha)} \left( (2N-3)N + 1 - \frac{(2N-1-2N)2}{N} \left( \frac{v_B}{2} \right)^{\frac{1-\alpha}{\alpha}} \right)^{N+1} - \frac{(2N-1)(N^2-1)}{2}$$

Hence $\epsilon^{NR} > \epsilon^{co}$. For $c > \frac{2\alpha}{(N+1)} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} > \frac{2\alpha}{(N+1)} \left( \frac{2}{2N} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{1-\alpha}{\alpha} \right)^{\frac{1-\alpha}{\alpha}}$, so that we know from Proposition 4 that $e_r = e_{nr}^{NR} > \epsilon^{co}$. ■

Proof of Proposition 7. a) Since $e_{nr}^{NR} < e_{nr}$ and $e_r^{NR} = e_r$, we get that $e_{nr}^{NR} - e_{nr}^{NR} < e_{nr} - e_r$. Hence for $c < c_3$, $e_{nr}^{NR} < e_r^{NR}$ whereas from propositions 2 and 4 $e_r^{NR} > e_r$. ■

Proof of Proposition 8. From Proposition 4, if $c > \frac{2\alpha}{(N+1)} \left( \frac{2}{v_B} \right)^{\frac{1-\alpha}{\alpha}} \left( \frac{2N-1}{2N} \right)^{\frac{1-\alpha}{\alpha}}$, $e_{nr}^{NR} > e_r^{NR} > \epsilon^{co}$. Since $\epsilon^{co}$ represents the maximum of the total surplus, total surplus is decreasing for all $\epsilon > \epsilon^{co}$. Thus surplus is higher when effort is lower, i.e. surplus with $e_r^{NR}$ is higher than surplus under $e_r^{NR}$.

■

References


