



# JENA ECONOMIC RESEARCH PAPERS



# 2010 – 034

## **Quality Competition or Quality Cooperation? License-Type and the Strategic Nature of Open Source vs. Closed Source Business Models**

by

**Sebastian von Engelhardt**

[www.jenecon.de](http://www.jenecon.de)

ISSN 1864-7057

The JENA ECONOMIC RESEARCH PAPERS is a joint publication of the Friedrich Schiller University and the Max Planck Institute of Economics, Jena, Germany. For editorial correspondence please contact [markus.pasche@uni-jena.de](mailto:markus.pasche@uni-jena.de).

Impressum:

Friedrich Schiller University Jena  
Carl-Zeiss-Str. 3  
D-07743 Jena  
[www.uni-jena.de](http://www.uni-jena.de)

Max Planck Institute of Economics  
Kahlaische Str. 10  
D-07745 Jena  
[www.econ.mpg.de](http://www.econ.mpg.de)

© by the author.

# Quality Competition or Quality Cooperation?\*

License-Type and the Strategic Nature of Open Source vs. Closed Source Business Models

S. v. Engelhardt\*\*

## Abstract

In the ICT sector, product-software is an important factor for the quality of the products (e.g. cell phones). In this context, open source software enables firms to avoid quality competition as they can cooperate on quality without an explicit contract. The economics of open source (OS) versus closed source (CS) business models are analyzed in a general two-stage model that combines aspects of non-cooperative R&D with the theory of differentiated oligopolies: In stage one, firms develop software, either as OS or CS, or as an OS-CS-mix if the license allows. In stage two, firms bundle this with complementary products and compete à la Cournot. The model allows for horizontal product differentiation in stage two. The findings are: 1.) While CS-decisions are always strategic substitutes, OS-decisions can be strategic complements. Furthermore, CS is a strategic substitute to OS and vice versa. 2.) The type of OS-license plays a crucial role: only if the license prohibits a direct OS-CS code mix (like the GPL), then Nash-equilibria with firms producing OS code exist for all parameters. 3.) In the equilibrium of a mixed industry with restricted licenses, OS-firms offer lower quality than their CS-rivals.

**Keywords:** open source, commercial open source, Cournot, R&D

**JEL:** D43, L17, O34

---

\*I am indebted to Steve Maurer (UC Berkely) for valuable discussions and suggestions. Furthermore, I would like to thank Georg von Graevenitz (LMU Munich), Markus Pasche (FSU Jena), the participants of the 'Augustin Cournot Doctoral Days' 2009, and the participants of the 'European Summer School on Industrial Dynamics' 2008, as well as the members of the DFG Graduate School 'The Economics of Innovative Change' in Jena for valuable comments and suggestions on earlier versions of this paper. Financial support from the **KLAUS TSCHIRA FOUNDATION** is gratefully acknowledged.

\*\*Friedrich-Schiller-University Jena, Department of Economics and Business Administration, Carl-Zeiss-Str. 3, D-07743 Jena. E-mail: [Sebastian.Engelhardt@wiwi.uni-jena.de](mailto:Sebastian.Engelhardt@wiwi.uni-jena.de)

## Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Open Source Business Models . . . . .	1
1.2. Previous Economic Research . . . . .	2
1.3. The Paper at Hand . . . . .	5
<b>2. Firms and Closed Source vs. Open Source Business Models:     The Basic Model Setup</b>	<b>7</b>
<b>3. Stage Two: Quantity Decisions</b>	<b>9</b>
<b>4. Stage One: Quality Decisions</b>	<b>10</b>
4.1. Liberal vs. Restricted OS-Licenses . . . . .	12
4.2. The Strategic Nature of CS-Decisions and OS-Decisions . . . . .	14
4.3. OS and CS in Case of Liberal Licenses . . . . .	16
4.4. OS and CS in Case of Restricted Licenses . . . . .	19
<b>5. Mixed Industries</b>	<b>25</b>
5.1. Liberal Licenses: OS and CS Development . . . . .	26
5.2. Restricted Licenses: Equilibrium of OS- and CS-Firms . . . . .	27
<b>6. Summary and Outlook</b>	<b>31</b>
<b>References</b>	<b>34</b>
<b>A. Appendix</b>	<b>39</b>

## 1. Introduction

Almost all of today's high tech products are computerized. While this is most obviously true for applications software (e.g. games), the point increasingly extends to hardware like cell phones and DVD players. In these industries, a product's quality—and hence consumer appeal—often depends sensitively on the software it contains. Before the 1990s, companies usually developed this as 'proprietary' software in-house. Since then, however, companies have increasingly turned to shared 'open source' code instead. So the ICT sector is characterized by the co-existence of open source (OS) and closed source (CS) software, the latter also called proprietary software. In the case of OS, the source code—i.e. the human-readable recipe of a software program—is 'open' (disclosed). This means that everybody has access to the software and its source code and the right to read, modify, improve, redistribute and use it. This principle of openness is codified in the copyright based OS licenses. Thus, OS software appears to be a case of a "private provision of a public good" (Johnson, 2002). Furthermore, the OS principle—especially in the context with profit-seeking firms—seems to represent a "new intellectual property paradigm" (Maurer and Scotchmer, 2006). So, some authors discuss the possibility to implement the 'open source' paradigm in further industries that are based on digital goods, i.e. "payoff-relevant bitstring[s]" (Quah, 2003), like DNA sequences (open source biology/biopharmaceutical/biotechnology etc., see Allarakhia et al., 2010; Bertacchini, 2008; Henkel and Maurer, 2007; Hope; Maurer, 2008; Pénin and Wack, 2008; Roosendaal, 2007).

### 1.1. Open Source Business Models

OS software is developed by a 'community' that consists of thousands of volunteers who develop software often without direct monetary reward. But more and more firms engage in the OS development, hence pay programmer to develop OS code. Profit-seeking firms like IBM, Motorola, Nokia, Novell, Panasonic, Philips, Red Hat, Sony, Sun Microsystems, as well as many small and medium sized enterprises use *OS business models*: As the OS-code itself can not be a profit center, OS business models are based on selling complementary products (Maurer and Scotchmer, 2006, p 289, 290ff). These

complements can be hardware like servers or cell phones, premium versions of the software, or different kinds of service like maintenance etc.<sup>1</sup> Unlike traditional joint venture partners, OS collaborators have no formal obligation to contribute any particular level of effort to these projects. Instead, companies must continuously balance the cost-savings from shared code development against the risk that they will make their competitors' products more desirable. This is true for all kinds of OS business models, not only in the context of software.

## 1.2. Previous Economic Research

One branch focuses on the incentives of the community members. Schiff (2002) provides an overview of early contributions regarding the question why "should thousands of top-notch programmers contribute freely to the provision of a public good?" (Lerner and Tirole, 2002). A prominent explanation refers to extrinsic motivations, namely to the acquisition of reputation-signals (Lerner and Tirole, 2000), but intrinsic motives also play a role. An empirical study can be found e.g. in Ghosh et al. (2002), see also the overviews in Rossi (2006) and David and Shapiro (2008). Institutional aspects, like licenses, governance of OS projects etc., were brought into focus by e.g. Bessen (2006); Brand and Schmid (2005); von Engelhardt and Freytag (2010); von Engelhardt (2008b); Gehring (2006); Weber (2004).

---

<sup>1</sup> Linux is used for several devices as *embedded software*, e.g. Amazon's Kindle, Cisco's MDS and Nexus data switches, Linksys' WRT54G W-LAN router, numerous Motorola, Nokia, and Panasonic mobile phones, Philips' LPC3180 microcontroller, TomTom's GPS navigation systems, various LG Panasonic, Samsung, and Sony LCD and plasma televisions. The most recent example of embedded OS is Android, a software stack (operating system, middleware and key applications) for mobile devices. Acer, Barnes & Noble, Dell, HTC Corporation/Google, Lenovo, LG, Motorola, Samsung, and Sony Ericsson all manufacture and sell products that come preinstalled with Android. Red Hat, Novell's SUSE and other Linux-*distributors* collect and optimize given OS software (ready-to-install 'distributions'), bundle this with further CS (for "Enterprise class" premium versions) and offer further services like support and maintenance. Most Web servers are driven by an OS "Lamp Stack" software suite that includes a Linux operating system, Apache Web server, MySQL database, and PHP/Perl/Python programming languages. Development is supported by corporations like Novell, IBM, Oracle, and Borland who then bundle LAMP with their proprietary hardware and software. Small web developers also use LAMP in their businesses and contribute code back to the project.

Some research on open innovation and user innovation focus on OS (von Hippel, 2005; von Hippel and von Krogh, 2003). The existence of non-commercial OS software has an impact on competition and adoption processes in software markets, analyzed e.g. by Casadesus-Masanell and Ghemawat (2006); Economides and Katsamakas (2006); Bitzer (2004) or Berends and van Wegberg (2000).

The increasing number of firms involved in OS inspired a recent branch of research focusing on the role and incentives of *profit-oriented firms doing OS*. Contrary to the rich empirical research (among many others Dahlander and Wallin, 2006; Fosfuri et al., 2008, 2005; Harison and Cowan, 2004; Harison and Koski, 2010; Lerner et al., 2006; Rossi and Bonaccorsi, 2006), there is less theoretical work on OS-business models, mostly limited to duopoly cases:

Baake and Wichmann (2004) analyze a duopoly-model where firms that can publish parts of their software as OS. Publishing code leads to positive spillovers i.e. reduce the firms' coding costs, but induce higher coding expenditures and thus decreases the firms' profits if their programs are substitutes. Additionally, it encourages entry and increases the expenditures required to deter entry. They find that both firms invest in OS to increase the quality and profitability of their respective CS products. Where the CS companies compete, however, each company must also increase the quality of its CS product to retain its customers. This effect is even stronger when the CS products compete with OS code and/or companies deliberately keep CS quality (and development costs) high to deter entry. Though intriguing, these results are limited to the duopoly case. Baake and Wichmann also make the very special assumption that OS costs rise faster than CS costs. Verani (2006) presents a Bertrand-duopoly model in which companies invest in either OS or CS software and then build products that use it. She finds that firms invest more when their products are substitutes, and that this effort is greater when OS software is used. Schmidtke (2006) analyzes OS business models in a non-differentiated Cournot oligopoly. Firms produce a homogeneous private good (e.g. a computer server) and invest in the quality of a homogeneous public good (OS software). He finds that the increasing number of firms in the market increases welfare, while the effects on each firm's private production and OS development, prices, and profits depend on the slope of the marginal costs of software development.

Henkel's 2006 "Jukebox Mode of Innovation" is a duopoly model to explore the case of embedded Linux. Crucially, Henkel assumes that all technologies are developed in-house without shared production of any kind; firms can, however, share costs by disclosing completed technologies to one another. Given this set-up, Henkel finds that each OS firm concentrates on developing whichever technology is most valuable to its business and copies the other technology from its rival. This creates a dynamic in which each company specializes in and controls the technology it values most so that total industry technology spending is biased upward. Henkel finds that OS industries deliver more technology and higher profits provided that firms do not compete too strongly with one another. However, these advantages disappear where both firms' products receive the same quality-increment from each technology. In this case, OS firms are reluctant to make their competitors stronger and therefore invest less than CS firms. Furthermore, firms are most likely to choose OS business models when competition is low, and each firm's technology needs are different.

Finally, two recent papers provide models in the tradition of Hotelling's model: In both models, there is a continuum of consumers, who differ in their valuations of the available products (heterogeneity in tastes). Furthermore, each consumer buys only *one* package (bundle), or nothing. In Casadesus-Masanell and Llanes (2009) consumers consume software and a complementary service. The software is further segmented into (a) a core program which consumers can use as a free-standing unit, and (b) extensions which are valueless without the core unit. They then examine when firms decide whether one or both software components should be developed as OS or CS, given three cases: monopoly, a firm vs. non-profit OS project, and duopoly. They find, inter alia, that firms are more willing to open modules when (a) consumer demand for the complementary good is strong, and (b) the quality of OS software is boosted by exogenous user innovation at no cost. Llanes and de Elejalde (2009) consider a model in which each firm sells packages consisting of a primary good (which can be OS or CS) and a complementary private good. Consumers have idiosyncratic preferences so that they usually favor one firm's private good over others. However, rival firms can overcome this preference by investing in a technology that simultaneously increases the quality of both the primary good and also the complement. Specifically, Llanes and DeElejalde analyze

a two stage model in which a predetermined number of firms (a) decide whether to produce OS or CS in the primary good, and then (b) simultaneously decide the quality/price of the bundle they will offer to consumers. They find that when most of the bundle's value comes from the primary good OS firms find it hard to appropriate profits from their investment in an open complement. This leads to outcomes in which a small number of firms choose CS and capture most of the market by delivering high quality code; the other firms become OS and deliver comparatively low quality code at a low price. However, this situation changes where consumers value the complement roughly as much as the primary. In this case, the cost advantage of code-sharing dominates so that all firms choose to become OS even though a hypothetical CS firm would produce higher quality software. This (theoretical) CS quality advantage reflects OS firms' limited ability to recover quality gains from consumers. The advantage disappears in cases where most of the bundle's value comes from the complementary.

So, regarding the role of profit-oriented players, there is still some lack of a general but simple theoretical analysis of the necessary and sufficient conditions for OS-production by profit seeking firms, and for the coexistence of firms with CS- versus OS-based business models (mixed industry).

### **1.3. The Paper at Hand**

The paper aims to fill this gap. We will analyze in a general way the economics of OS vs. CS business models in terms of strategical aspects, including the role of the OS-license type and of the non-commercial community (the hobbyists), and the resulting industry equilibrium. The model is based on simple linear demand, and as a general oligopoly model it covers a wide scope of possible situations, ranging e.g. from duopoly with completely separated markets to perfect competition (infinity number of firms, perfect substitutes). Although the model is inspired by, and refers to, the case of software, its application is not limited. It analyzes the 'economics of commercial open source' in a general way, and can thus be applied to other examples like 'open source biology' etc.

As already mentioned, OS business models combine commonly developed code with individually produced and sold complements. The underlying strategic logic is of a R&D cooperation *without* an explicit contract.



In order to analyze the economics of such contract-free collaborations, we develop an oligopoly model, where firms can do OS, or CS, or both. So, the model is in tradition of (non-)cooperative R&D models, most prominently represented by [d'Aspremont and Jacquemin \(1988\)](#). Whereas the literature on (non-)cooperative R&D (e.g. see [de Bondt \(1997\)](#) for an overview) analyzes R&D with *given* spillovers, an OS vs. CS decision of the firm determines whether 'spillovers' exists or not. Note that regardless whether CS or OS (or a mix) is chosen, it is always a case of non-cooperative action.

Firms doing such contract-free OS collaborations can act in different types of markets. For example, Linux is a platform upon which firms in several markets build their business models. Therefore we allow for horizontal product differentiation and the model is based on the theory of differentiated oligopoly/duopoly like proposed by [Dixit \(1979\)](#) and further developed e.g. by [Singh and Vives \(1984\)](#) or [Häckner \(2000\)](#).

We will also analyze the impact of an important institution on the outcome of the game: the type of OS license. Different OS projects use different types of licenses. These types differ with respect to whether the use of the code is restricted or how the use is restricted respectively. For example, so-called public licenses, like the BSD<sup>2</sup> license, do not restrict the use of the software and the source code in any way and thus allows OS-CS mixed code. We will call such licenses liberal licenses. Other licenses are more restrictive. One famous example is the GPL<sup>3</sup>. This license claims, that any further developed software as well any derived work must be licensed as a whole under the same license. This clause wants to make sure, that OS code stays "open", and is also known as the "copyleft" principle. We will call such licenses restricted licenses as they prohibit OS-CS mixed code.

The remainder of this paper proceeds as follows. Section 2 introduces the basic model setup. As the model is a two-stage game it will be solved by backward induction, i.e. section 3 provides the solution of stage two, while section 4 analyzes the OS and CS decision of firms (stage one) given liberal and restricted OS licenses. Section 5 deals with the equilibrium ratio of CS- and OS-firms, assumed free entry and exit (mixed industry). Section 6 provides a summary of the findings and an outlook.

---

<sup>2</sup>BSD stands for Berkeley Software Distribution

<sup>3</sup>The GPL (GNU General Public License) is the most popular OS license.

## 2. Firms and Closed Source vs. Open Source Business Models: The Basic Model Setup

In many markets, software is sold *bundled* with complementary products like service (maintenance, individualizing) or hardware. Here firms make money with business models that are based on either CS or OS software, or software consisting of mixed CS-OS code. The basic principle of open source business models is therefore to develop OS code together with others, and then make money with selling the (bundled) complements. This is the combination of a public good with a private good, or: a combination of non-cooperative R&D (OS-firms do not have an explicit contract with each other) with oligopolistic competition where products can be vertically (quality) and horizontally differentiated.

Therefore, let us consider a market with  $n \geq 2$  firms. One arbitrary firm is denoted by  $i$ , with  $i \in N = \{1, 2, \dots, n\}$ . Each firm  $i \in N$  produces quantities of a horizontal differentiated product, and develop complementary software which can be either OS or CS code, or, if the license allows mixing, an OS-CS code respectively. The software and the product are then sold as bundle  $q_i$ . Firm  $i$ 's software has a direct impact on the quality of the bundle  $q_i$ , i.e. differences in software lead to vertical product differentiation. We thus need a *utility function* that enables us to take into account horizontal and vertical product differentiation. There are two approaches: *One version* was proposed by Sutton (1997, p 618) and used e.g. by Symeonidis (2003) or Deroian and Gannon (2006):

$$V = \sum_{i=1}^n \left( q_i - \frac{q_i^2}{v_i} \right) - 2\gamma \sum_i \sum_{j < i} \frac{q_i q_j}{v_i v_j} + I. \quad (1)$$

The *second version* was introduced by Dixit (1979) for the duopoly case. Häckner (2000) and Hsu and Wang (2005) provide a generalized version of Dixit's utility function, i.e. the oligopoly-version:

$$U = \sum_{i=1}^n q_i \alpha_i - \frac{1}{2} \left( \sum_{i=1}^n q_i^2 + 2\gamma \sum_{i \neq j} q_i q_j \right) + I \quad (2)$$

In both functions,  $I$  is the composite good, with its price normalized to one. The parameter  $\gamma \in [0, 1]$  is an inverse measure of the horizontal product differentiation:  $\gamma$  indicates the (horizontal) substitutability between the different products, with  $\gamma = 1$  for perfect substitutes. The parameter  $\alpha_i$ ,  $v_i$  respectively, represent the quality level of the bundle, such that vertical product differentiation is expressed by different  $\alpha$ s, different  $v$ s respectively. Please notice that in our model setup both types of utility function lead to similar results (see p 9 footnote 6). We will use the *second* one, i.e. function (2).

In the following, we consider a two-stage game, that combines competition in quantities with quality-competition/cooperation: Firms decide about quantity and quality (via software). As we will show below, the decision to develop OS rather than CS code is a decision to cooperate rather than compete on quality.

1. In *stage one*, firms decide about their 'stage one'-software development. Hence, they choose their optimal amount of OS or CS. As this affects quality, stage one represents the *quality decision*. As already mentioned, we distinguish between different types of OSS licenses. We thus analyze stage one with *liberal* and with *restricted* licenses.
2. In *stage two* oligopolistic competition takes place. The firms produce their 'stage two'-products,<sup>4</sup> bundle this with the complementary 'stage one'-software, and compete à la Cournot. This means, in stage two profit-maximizing *quantities* are defined.

The game will be solved by backward induction. Hence, in the next section we start with stage two. In section 4 we then analyze stage one.

---

<sup>4</sup> Please note that the 'stage two'-product *can* be (closed source) software. The bundle then consists of 'stage one'-software plus 'stage two'-software, i.e. is a kind of 'Premium Version' of the 'stage one'-software. However, we will refer to the software developed in stage one as 'stage one'-software, or 'software'. And we will call the product produced in stage two as 'stage two'-product or 'product', whether it is software or not.

### 3. Stage Two: Quantity Decisions

In the stage two, the oligopolistic quantity competition with the horizontally and vertically differentiated product-bundles takes place. Without loss of generality, we normalize the marginal costs of the ‘stage two’-product to zero. Fixed costs of ‘stage two’-products are given by  $C$ .<sup>5</sup> Hence firms  $\max_{q_i} \pi_i = p_i q_i - c_i - C$ , where  $p_i = \alpha_i - q_i - \gamma \sum_{j \neq i} q_j$  is the inverse demand function derived from the utility function (2), and  $c_i$  are the software development costs determined in stage one.

The resulting equilibrium prices and quantities of this differentiated oligopoly are given by:

$$p_i = q_i = \frac{\alpha_i + \frac{\gamma}{(2-\gamma)} \sum_{j \neq i} (\alpha_i - \alpha_j)}{2 + \gamma(n-1)} = \frac{\alpha_i + \theta \sum_{j \neq i} (\alpha_i - \alpha_j)}{h}. \quad (3)$$

The resulting revenue function, simply the square of the above expression, has strong similarities to the revenue function one would achieve using the utility function  $V$  (function (1), p 7).<sup>6</sup>

We have introduced  $\theta$  and  $h$  because this is convenient for interpreting the results later on. As already mentioned the model combines quantity competition (Cournot) with quality competition/cooperation. The measure  $h = 2 + \gamma(n-1)$  indicates the degree of *quantity competition*, and depends on the number of competitors weighted by the degree of substitution. (It is simply the denominator one can find in any differentiated Cournot model.) Second,  $\theta = \gamma/(2-\gamma)$  indicates how much differences in quality affect firm’s

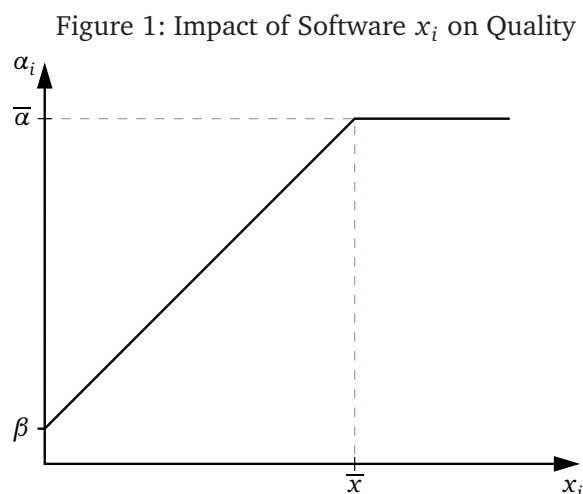
<sup>5</sup>If the ‘stage two’-product is also software,  $C$  represents the software development costs, i.e. the so-called first copy costs of this ‘stage-two’-software. See also footnote 4, p 8

<sup>6</sup>On p 633 Deroian and Gannon (2006) provide a revenue function based on the utility function (1), given by  $2S \left( u_i (\sigma(n-2) + 4) - \sigma \sum_{j \neq i} u_j \right)^2 (4 - \sigma)^{-2} (\sigma(n-1) + 4)^{-2}$ . Notice, that we stick to *their* notation. In order to to obtain *our* notation, we have to replace  $\sigma$  by  $2\gamma$  and replace  $u_i$  by  $\alpha_i$ . We finally obtain an equation that differs from the revenue function of our model only by the term  $\frac{1}{2}S$ . Deroian & Gannon denote with  $S$  the number of consumers. Thus  $S$  represents the size of the market, and the difference is in the *level* of the returns only. (If we normalize  $S = 2$ , the two revenue functions become equal.) The similarities between outcomes of models based on the two different types of utility functions are also mentioned by other authors, see for example Symeonidis (2003, p 42, Appendix A).

revenue. Therefore  $\theta$  indicates degree of *quality competition* and hence measures the incentive to compete rather than to cooperate on quality. In other words:  $h$  and  $\theta$  enables to separates the effects of a change of  $\gamma$ . If  $d\gamma > 0$  then this (a) has a negative impact on revenues, as a firm's revenue c. p. decrease with an increase in  $h$ , (b) has a positive impact on a firm's revenue given this firm has an advantage in quality.

#### 4. Stage One: Quality Decisions

Firm  $i$ 's 'stage one'-software has a direct impact on the quality of the bundle, i.e. on  $\alpha_i$ . Let  $\alpha_i = \beta + x_i$ , with  $x_i$  as the software firm  $i$  can use for its bundle. The parameter  $\beta > 0$  catches the demand-relevant effects of the quality of the 'stage two'-product. For the sake of simplicity, the impact of 'stage one'-software ( $x_i$ ) on  $\alpha_i$  is modeled linearly with an upper boundary  $\alpha_i \in [0, \bar{\alpha}]$  that yields a cutoff  $\bar{x}$ , see figure 1. This is done in order to avoid that—just as an artefact of the model setup—software development can shift  $\alpha_i$ , and hence demand, towards infinity.<sup>7</sup>



<sup>7</sup>The used quality function can be interpreted as being an approximation of a logistic function. As we assume, that firms never develop more that  $\bar{x}$ , the quality function can also be interpreted as being an approximation of an inverse U-shaped function.

In line with the literature, we assume that software development faces increasing marginal costs, as the costs of software development are driven by the rising complexity of the code. Software is in some sense a ‘logical machine’, and the more sophisticated a software program becomes, the more complex the whole system gets. Modern software development is far more than just writing lines of code, and also consists of finding and fixing mistakes (so-called bugs). Hence the costs of software development consists of code-writing costs but also of bug-avoiding costs (ex-post designing, coordination and control), bug-finding and bug-fixing costs. As complexity of modern software rises non-linear, this yields rising marginal costs of software development (see also von Engelhardt, 2008a, p 14 ff).

This fact is approximated by a quadratic cost function. Let  $\phi > 0$  denote the slope of the marginal costs. Given  $x$  is a software,<sup>8</sup> then the total costs are given by  $c(x) = \frac{1}{2} \cdot \phi \cdot x^2$ . The total costs are independent of whether the code is OS or CS. OS has thus no inherent cost advantage over CS except to the extent that it allows collaborating members to share cost. If a firm develops CS code then it bears the total costs of this  $x_i^{cs}$ , i.e. the firm’s costs are given by  $c_i(x_i^{cs}) = \frac{1}{2}\phi x_i^{cs2}$ . But if a firm develops OS code then there is the cost-sharing effect, as the code is developed collaboratively with other OS-developing firms or members of the community. Thus depending on its own contribution ( $x_i^{os}$ ), the firm bears only a fraction  $k$  of the total OS-costs. Let  $X^{os}$  denote the total OS code. As the firm bears only that fraction that is caused by its own development  $x_i^{os}$ , i.e.  $k = x_i^{os}/X^{os}$ , this yields  $c_i(x_i^{os}) = k \cdot c(X^{os}) = \frac{1}{2}\phi x_i^{os} X^{os}$ . Hence  $(1 - k) \cdot c(X^{os})$  is born by the remaining OS-developing firms and members of the community. Of course, the latter holds independently whether the firm develops *only* OS or *mix* OS and CS code. Taking the above aspects together, the cost function of a firm  $i$  that does OS and CS is given by  $c_i = \frac{1}{2}\phi (x_i^{os} + x_i^{cs})^2 + \frac{1}{2}\phi X_{-i}^{os} (x_i^{os} + 2x_i^{cs})$ , with  $X_{-i}^{os} = X^{os} - x_i^{os}$ .

In summary it can be said, that software development has benefits (positive impact on  $\alpha_i$ , and hence on revenues) as well as its development costs. Clearly, if a firm develops OS rather CS code, then this affects also the costs- and benefits-aspect. The OS principle is a collaborative way of devel-

---

<sup>8</sup>Note that this notation subsumes qualitative and quantitative aspects of the software, hence a higher value of  $x$  can indicate more functions as well as ‘better’ functions

oping *and* using coded, digital goods. While the principle of closed source is based on private costs and benefits, the open source principle implies a cost and benefit sharing:

- Costs: OS code is *jointly developed* code. This implies cost sharing, including the complexity costs. (This implies that marginal costs of OS development of a firm are smaller than the marginal costs of CS.)
- Benefits: OS code is *jointly used* code. This implies benefit sharing, as all firms who use the OS code benefit from its impact on quality. Developing OS code is to cooperate rather than to compete on quality. For CS code the opposite holds.

These two aspects determine whether OS or CS is more attractive, and is strongly influenced by an important institution: the type of OS-license. It will turn out, that this institutional difference matters, especially in cases where firms are the *only* potential OS contributors, hence where no (non-commercial) OS-community exists.

#### 4.1. Liberal vs. Restricted OS-Licenses

We distinguish two types of OS-licenses: unrestricted—or: liberal—ones, and restricted ones. While a *liberal license* permits to mix OS with CS code, a *restricted license* prohibits any mixing of OS and CS code at the level of ‘stage one’-software:<sup>9</sup>

- In case of *liberal licenses* firms can mix OS and CS code. Therefore  $\alpha_i$  is given by

$$\alpha_i = \beta + x_i^{\text{CS}} + x_i^{\text{OS}} + X_{-i}^{\text{OS}} \quad (4)$$

and the costs  $c_i$  are

$$c_i = \frac{1}{2}\phi (x_i^{\text{OS}} + x_i^{\text{CS}})^2 + \frac{1}{2}\phi X_{-i}^{\text{OS}} (x_i^{\text{OS}} + 2x_i^{\text{CS}}). \quad (5)$$

---

<sup>9</sup> The prohibition to mix affects stage one! Bundling OS software with ‘stage two’-CS software is possible.

- In case of *restricted license* firms have to *choose* between OS and CS. This separates firms into OS-firms and CS-firms. We will denote the number of OS-firms by  $z$  and the number of CS-firms by  $r$ , such that  $r + z = n$  (The sets of firms are denoted by capital letters:  $Z \cup R = N$  with  $Z \cap R = \emptyset$ ). This yields

$$\alpha_i = \begin{cases} \beta + x_i^{\text{CS}} & \text{if } i \in R \text{ (CSS-firm)} \\ \beta + x_i^{\text{OS}} + X_{-i}^{\text{OS}} & \text{if } i \in Z \text{ (OSS-firm)}, \end{cases} \quad (6)$$

and for the cost function

$$c_i = \begin{cases} \frac{1}{2} \phi x_i^{\text{CS}2} & \text{if } i \in R \text{ (CSS-firm)} \\ \frac{1}{2} \phi x_i^{\text{OS}} (x_i^{\text{OS}} + X_{-i}^{\text{OS}}) & \text{if } i \in Z \text{ (OSS-firm)}. \end{cases} \quad (7)$$

In stage one the firms maximize the profits  $\pi_i = p_i \cdot q_i - c_i - C$  with respect to  $x_i^{\text{OS}}$  and/or  $x_i^{\text{CS}}$ .

In case of *liberal licenses* firms decide about their optimal OS and CS code. The resulting reaction functions for the OS and CS output of firm  $i$  are

$$R_i^{\text{CS}} = \frac{(1 + (n - 1) \theta) (\beta - \theta X_{-i}^{\text{CS}}) - \left( \frac{1}{2} \phi h^2 - (1 + (n - 1) \theta) \right) (x_i^{\text{OS}} + X_{-i}^{\text{OS}})}{\frac{1}{2} \phi h^2 - (1 + (n - 1) \theta)^2}, \quad (8)$$

$$R_i^{\text{OS}} = \frac{\beta - \theta X_{-i}^{\text{CS}} - \left( \frac{1}{2} \phi h^2 - (1 + (n - 1) \theta) \right) x_i^{\text{CS}} - \left( \frac{1}{4} \phi h^2 - 1 \right) X_{-i}^{\text{OS}}}{\frac{1}{2} \phi h^2 - 1} \quad (9)$$

Because of the liberal license firm  $i$ 's OS and CS are substitutes:  $\partial R_i^{\text{CS}} / \partial x_i^{\text{OS}} < 0$  and  $\partial R_i^{\text{OS}} / \partial x_i^{\text{CS}} < 0$ , with  $|\partial R_i^{\text{CS}} / \partial x_i^{\text{OS}}| \geq |\partial R_i^{\text{OS}} / \partial x_i^{\text{CS}}|$ .

In the case of *restricted license* we obtain the following reaction functions for the OS-firms and the CS-firms:

$$R_{i \in R}^{\text{CS}} = \frac{(1 + (n - 1) \theta) \left( \beta - \theta \sum_{j \neq i} x_j^{\text{CS}} - \theta z X^{\text{OS}} \right)}{\frac{1}{2} \phi h^2 - (1 + (n - 1) \theta)^2}, \quad (10)$$

$$R_{i \in Z}^{\text{OS}} = \frac{(1 + r \theta) (\beta - \theta r x^{\text{CS}}) - \left( \frac{1}{4} \phi h^2 - (1 + r \theta)^2 \right) X_{-i}^{\text{OS}}}{\frac{1}{2} \phi h^2 - (1 + r \theta)^2}. \quad (11)$$



Regarding the second order conditions (SOCs), the following holds: For CS the SOC  $\phi > \phi_{\text{soc}}^{\text{CS}}$  is defined by  $\phi_{\text{soc}}^{\text{CS}} = 2(1+(n-1)\theta)^2/h^2$  in case of liberal as well as restricted licenses. The SOC of OS,  $\phi > \phi_{\text{soc}}^{\text{OS}}$ , is defined by  $\phi_{\text{soc}}^{\text{OS}} = 2/h^2$  for liberal licenses, and by  $\phi_{\text{soc}}^{\text{OS}} = 2(1+r\theta)^2/h^2$  for restricted licenses. Furthermore,  $\phi_{\text{soc}}^{\text{CS}} > \phi_{\text{soc}}^{\text{OS}}$ . As usual, in the following it is assumed that each second order condition is fulfilled. (If not, “more code” is always better. This would yield that firms develop the cutoff  $\bar{x}$ )

#### 4.2. The Strategic Nature of CS-Decisions and OS-Decisions

Because of the respective nature of OS and CS, both types of software development differ in terms of strategic complements or substitutes. The terms of strategic complements or substitutes were originally introduced by [Bulow et al. \(1985\)](#). Decisions of players are strategic substitutes if they mutually cut back one another. Decisions are strategic complements if the reverse is true. We will express this with the elasticities.

CS-decisions are strategic substitutes as optimal CS development always decreases when CS code of the other firms increases:  $\mathcal{E}_{\text{CS}}^{\text{CS}} = \partial x_i^{\text{CS}} / \partial x_j^{\text{CS}} \cdot x_j^{\text{CS}} / x_i^{\text{CS}} < 0$ . But optimal OS reacts on other players’ OS either in a positive or a negative way. If the slope of the marginal costs  $\phi$  does not exceed a certain threshold, the decisions about OS are decisions in strategic substitutes, i.e.  $\mathcal{E}_{\text{OS}}^{\text{OS}} = \partial x_i^{\text{OS}} / \partial x_j^{\text{OS}} \cdot x_j^{\text{OS}} / x_i^{\text{OS}} > 0$ . Of course this also holds for OS-code contributed by the non-commercially oriented members of the OS-community. We will call this community-code and denote it with  $x_{\text{nc}}^{\text{OS}}$ , with “nc” for “non-commercial”. Of course,  $\mathcal{E}_{\text{nc}}^{\text{OS}} = \partial x_{\text{nc}}^{\text{OS}} / \partial x_{\text{nc}}^{\text{OS}} \cdot x_{\text{nc}}^{\text{OS}} / x_{\text{nc}}^{\text{OS}}$  can also be greater than one. Finally, the positive feedbacks between the OS-players can be very strong: If  $\partial R_i^{\text{OS}} / \partial x_j^{\text{OS}} > 1$  then this implies a symmetric Nash-equilibrium where the OS-developing firms together develop the cutoff. In particular:

**Proposition 4.1.** *CS-decisions are strategic substitutes:  $\mathcal{E}_{\text{CS}}^{\text{CS}} < 0$ .*

*Proof.* Given the second order condition is fulfilled, then it is true that  $\partial R_i^{\text{CS}} / \partial x_j^{\text{CS}} < 0$ , which also implies  $\mathcal{E}_{\text{CS}}^{\text{CS}} < 0$ .  $\square$

**Proposition 4.2.** *OS-decisions are strategic substitutes ( $\mathcal{E}_{\text{OS}}^{\text{OS}} < 0$ ,  $\mathcal{E}_{\text{nc}}^{\text{OS}} < 0$ ) if and only if  $\phi > 2 \cdot \phi_{\text{soc}}^{\text{OS}}$ , otherwise strategic complements ( $\mathcal{E}_{\text{OS}}^{\text{OS}} > 0$ ,  $\mathcal{E}_{\text{nc}}^{\text{OS}} > 0$ ).*

*Proof.* If the second order condition is fulfilled,  $\partial R_i^{os}/\partial x_j^{os} < 0$  and  $\partial R_i^{os}/\partial x_{nc}^{os} < 0$  is true only for  $\phi > 2\phi_{soc}^{os}$ , otherwise is  $\partial R_i^{os}/\partial x_j^{os} > 0$  and  $\partial R_i^{os}/\partial x_{nc}^{os} > 0$ . Thus for  $\phi > 2 \cdot \phi_{soc}^{os}$  is  $\mathcal{E}_{os}^{os} < 0$ ,  $\mathcal{E}_{nc}^{os} < 0$ , otherwise  $\mathcal{E}_{os}^{os} > 0$ ,  $\mathcal{E}_{nc}^{os} > 0$ .  $\square$

**Proposition 4.3.** For  $\phi < \phi_{\bar{x}}^{os}$  there exists a symmetric Nash-equilibrium of OS-development such that  $X^{os} = \bar{x}$ .

*Proof.* From  $\partial R_i^{os}/\partial x_j^{os} \stackrel{!}{=} 1$  one obtains the boundary  $\phi_{\bar{x}}^{os}$ , which is given (a) by  $\phi_{\bar{x}}^{os} = n/(n+1) \cdot 2\phi_{soc}^{os}$  in the case of liberal licenses, and (b) by  $\phi_{\bar{x}}^{os} = z/(z+1) \cdot 2\phi_{soc}^{os}$  in the case of restricted licenses. For all  $\phi < \phi_{\bar{x}}^{os}$  it holds that  $\partial R_i^{os}/\partial x_j^{os} > 1$ . Thus with respect to OS the players react positive on each other with the factor greater than one. Put simply, the positive strategic interplay of OS is so strong that the players keep on mutually pushing up. This leads to the fact that the upper corner solution is always an equilibrium i.e. the OS-firms develop together  $X^{os} = \bar{x}$ . Symmetry implies that in this Nash-equilibrium each OS-firm develop the same fraction of  $\bar{x}$ . See also figure 7 in the Appendix  $\square$

Proposition 4.3 has the following intuition: If  $\phi < \phi_{soc}^{os}$  then developing more OS code is always better independent from what the other OS-players do. The reason is the relatively low slope of the marginal costs. For  $\phi_{soc}^{os} < \phi < \phi_{\bar{x}}^{os}$  developing more OS code is always better, because of what the other OS-players do. Thus the positive feedback among the players, the strong incentives to cooperate in OS shifts the boundary of ‘more OS code is always better’ upwards. Analyzing this boundary provides some insight in the nature of the strategic interaction of OS code by firms.  $\phi_{\bar{x}}^{os}$  is equal  $n/(n+1) \cdot 4/h^2$  or  $z/(z+1) \cdot 4(1+r\theta)^2/h^2$ , depending on the type of license. Recall that  $h = 2 + (n-1)\gamma$  and  $\theta = \gamma/(2-\gamma)$ , with the latter indicating the incentives to compete on quality. If licenses are restricted then the OS-firms cooperate on quality among each other, but compete on quality with the CS-firms. Therefore an increase of  $\gamma$ —which implies an increase of  $\theta$ —has a positive impact on the incentives to develop more OS software, namely via  $(1+r\theta)^2$ . At the same time,  $d\gamma > 0$  implies  $dh > 0$  and thus lowers the incentives to develop more OS simply because of the increased degree of competition. Also the number of firms, of OS-firms respectively, has a twofold and opposing impact on incentives to develop OS software. On the

one hand an increase in  $n$ , in  $z$  respectively, yields  $dh > 0$ . On the other hand, in order to reach a certain code-output, each firm has to contribute less if more firms collaborate. This is expressed by  $n/(n+1)$ , and  $z/(z+1)$  respectively.

However, in the following we will concentrate on situations where  $\phi$  is high enough to ensure that total OS code is not equal the cutoff. The reason for this is that if  $X^{\text{os}} = \bar{x}$  further model results like the equilibrium number of OS- and CS-firms is determined by this exogenous value, rather than by endogenous results.

### 4.3. OS and CS in Case of Liberal Licenses

In this section we analyze the case of *liberal licenses*. In this context, the community-code plays the following role: As liberal license allow for mixed OS-CS code, code from the non-commercial OS-community ( $x_{\text{nc}}^{\text{os}}$ ) directly impacts on both, OS as well as CS development by firms. If there is enough community-code, firms do not develop any further software as they get enough code “delivered for free”. With respect to firm-developed OS there is also a second effect of  $x_{\text{nc}}^{\text{os}}$ . If the firms’ markets are sufficiently separated, then firms will develop OS code anyway. But if  $\theta$  is higher, then there must be at least some  $x_{\text{nc}}^{\text{os}}$  to ensure that firms produce OS software. As we will derive below, in the case of  $x_{\text{nc}}^{\text{os}} = 0$  firms contribute OS code only if  $\theta < 1/(n+1)$ . The reason is that for  $\theta < 1/(n+1)$  quality competition measured by  $\theta$  is low. As the substitutability ( $\gamma$ ) is low, the firms’ markets are separated enough so that the firms have only low incentives to compete on quality and therefore cooperating on quality is more attractive. If  $\theta > 1/(n+1)$  the incentives to compete on quality are higher. To ensure OS contributions by firms also on this case, this incentives have to be overcompensated by  $x_{\text{nc}}^{\text{os}} > 0$ . With volunteers who contribute OS code, firms benefit more from developing OS than without them, as now the development costs are born jointly by the firms and the volunteers.<sup>10</sup> However, the positive effect of  $x_{\text{nc}}^{\text{os}}$  on the incentives to develop OS code has its limits. If  $\theta > 1/(n-1)$  then there is no firm-developed OS code at all. This means that in the case of liberal

<sup>10</sup>In some sense the non-commercial community is like a firm who develops code but do not compete at all with the other firms. This ‘firm’ has a  $\gamma$  equal zero, and thus a  $\theta$  equal zero. With this ‘firm’ the average  $\theta$  decreases such that it is again below the threshold.

licenses, if the number of firms is large, there are virtually no situations where firms produce OS code, no matter whether  $x_{nc}^{OS} > 0$  or not (see also section 5.1).

In detail the following propositions hold:

**Proposition 4.4.** *A symmetric Nash-equilibrium where the firms develop only CS ( $x^{CS*} > 0, x^{OS*} = 0$ ) exists if and only if the conditions  $x_{nc}^{OS} < \beta(1+(n-1)\theta)/\eta$  and  $x_{nc}^{OS} < \beta 2\theta(n-1)/\eta$  are fulfilled, with  $\eta = \frac{1}{2}\phi h^2 - (1 + (n-1)\theta)$ .*

*Proof.* If all firms develop *only* CS code, then it is  $x_i^{OS} = 0 \forall i \in N$  and thus the symmetric solution of the CS reaction functions—see (8)—is given by

$$x^{CS*} = \frac{\beta(1+(n-1)\theta)}{\frac{1}{2}\phi h^2 - (1+(n-1)\theta)} - x_{nc}^{OS}.$$

First,  $x^{CS*}$  is greater zero only if  $x_{nc}^{OS} < \beta(1+(n-1)\theta)/\eta$ .<sup>11</sup> Second, to ensure that  $(x^{OS} = 0, x^{CS} > 0)$  is an equilibrium, the optimal amount of OS must be zero. The zero of the symmetric solution obtained from the OS reaction function (9) with  $X_{-i}^{CS} = (n-1) \cdot x^{CS*}$  and  $x_i^{CS} = x^{CS*}$  delivers the boundary  $x_{nc}^{OS} = \beta 2\theta(n-1)/\eta$ . For values below this boundary optimal OS is zero. Taking the two conditions together we can conclude that only for  $x_{nc}^{OS}$  that fulfill  $x_{nc}^{OS} < \beta(1+(n-1)\theta)/\eta$  and  $x_{nc}^{OS} < \beta 2\theta(n-1)/\eta$  a Nash-equilibrium with  $x^{CS*} > 0, x^{OS*} = 0$  exists.  $\square$

**Proposition 4.5.** *A symmetric Nash-equilibrium where the firms develop only OS ( $x^{CS*} = 0, x^{OS*} > 0$ ) exists if and only if  $\beta((n+1)\theta-1)(n-1)/\eta < x_{nc}^{OS} < \beta/(\frac{1}{4}\phi h^2-1)$ , with  $\eta = \frac{1}{2}\phi h^2 - (1 + (n-1)\theta)$ .*

*Proof.* If all firms develop *only* OS, then it is  $x_i^{CS} = 0 \forall i \in N$  and thus the symmetric solution of the OS reaction functions—see (9)—is given by

$$x^{OS*} = \frac{\beta - \left(\frac{1}{4}\phi h^2 - 1\right) x_{nc}^{OS}}{\frac{1}{4}\phi h^2 + n \left(\frac{1}{4}\phi h^2 - 1\right)}.$$

<sup>11</sup>Otherwise firms do not develop CS as there is no incentive to do so because enough code is delivered by the non-commercial community.

First,  $x^{\text{OS}*}$  has to be greater zero. If  $\phi < 2\phi_{\text{soc}}^{\text{OS}}$  (strategic complements) then  $x^{\text{CS}*} > 0 \forall x_{\text{nc}}^{\text{OS}}$ . Otherwise  $x^{\text{OS}*} > 0$  only if  $x_{\text{nc}}^{\text{OS}} < \beta/(\frac{1}{4}\phi h^2 - 1)$ .<sup>12</sup> Second, to ensure that  $(x^{\text{OS}} > 0, x^{\text{CS}} = 0)$  is an equilibrium, it must be that the optimal amount of CS is zero. The zero of the symmetric solution obtained from the CS reaction function (8) with  $X_{-i}^{\text{OS}} = (n-1) \cdot x^{\text{OS}*} + x_{\text{nc}}^{\text{OS}}$  and  $x_i^{\text{OS}} = x^{\text{OS}*}$  delivers the boundary  $x_{\text{nc}}^{\text{OS}} = \beta((n+1)\theta - 1)(n-1)/\eta$ . Taking the two conditions together we obtain that only for  $\beta((n+1)\theta - 1)(n-1)/\eta < x_{\text{nc}}^{\text{OS}} < \beta/(\frac{1}{4}\phi h^2 - 1)$  a Nash-equilibrium with  $x^{\text{CS}*} = 0, x^{\text{OS}*} > 0$  exists.  $\square$

Notice that in both cases, code-output decreases when quantity competition ( $h$ ) becomes more intensive. In case of CS-only is  $\partial x^{\text{CS}}/\partial h < 0$  and in case of OS-only is  $\partial x^{\text{OS}}/\partial h < 0$ . The reason is that strong quantity competition limits the appropriability of quality investments in any case. Furthermore, while CS reacts to the degree of quality competition ( $\partial x^{\text{CS}}/\partial \theta > 0$ ), the OS of OS-only lacks of  $\theta$ . The reason is that in case of OS firms cooperate on quality: they avoid quality competition through code-sharing. Finally, the OS-function also reflects the cost-sharing aspect by  $(\frac{1}{4}\phi h^2 - 1)$ .

**Proposition 4.6.** *A symmetric Nash-equilibrium where the firms develop OS and CS ( $x^{\text{CS}*} > 0, x^{\text{OS}*} > 0$ ) exists if and only if  $\beta((n+1)\theta - 1)(n-1)/\eta < x_{\text{nc}}^{\text{OS}} < \beta 2\theta(n-1)/\eta$ , with  $\eta = \frac{1}{2}\phi h^2 - (1 + (n-1)\theta)$ .*

*Proof.* Optimal CS and OS are given by the reaction functions (8) and (9). We make use of the fact that, because of symmetry, in equilibrium  $\forall i: x_i^{\text{OS}} = x^{\text{OS}}$  and  $x_i^{\text{CS}} = x^{\text{CS}}$ . Reciprocal substitution leads to the solution for (8) and (9). The symmetric Nash-equilibrium where the firms develop OS and CS is given by

$$x^{\text{CS}*} = \frac{\beta(1 - (n+1)\theta)}{\frac{1}{2}\phi h^2 - (1 + (n-1)\theta)} + \frac{x_{\text{nc}}^{\text{OS}}}{n-1},$$

$$x^{\text{OS}*} = \frac{\beta 2\theta}{\frac{1}{2}\phi h^2 - (1 + (n-1)\theta)} - \frac{x_{\text{nc}}^{\text{OS}}}{n-1}.$$

From the two non-negativity constraints ( $x^{\text{CS}*} > 0$  and  $x^{\text{OS}*} > 0$ ) we directly derive the two conditions  $x_{\text{nc}}^{\text{OS}} > \beta((n+1)\theta - 1)(n-1)/\eta$  and  $x_{\text{nc}}^{\text{OS}} < \beta 2\theta(n-1)/\eta$ .  $\square$

<sup>12</sup>If  $x_{\text{nc}}^{\text{OS}}$  exceeds this level, then there is so much community code that firms do not have any incentives to develop any further code.

**Proposition 4.7.** *If  $\theta > 1/(n-1)$  and if there is software development by firms, then there is CS only.*

*Proof.* First, for  $\theta > 1/(n-1)$  it is true that  $\beta((n+1)\theta-1)(n-1)/\eta > \beta 2\theta(n-1)/\eta$ . Thus an equilibrium with  $x^{\text{CS}^*} > 0$ ,  $x^{\text{OS}^*} > 0$  can not exist (see proposition 4.6). Second, if  $\theta > 1/(n-1)$  then  $\beta((n+1)\theta-1)(n-1)/\eta > \beta/(\frac{1}{4}\phi h^2 - 1)$ , with  $\eta = \frac{1}{2}\phi h^2 - (1 + (n-1)\theta)$ . This yields that for  $\theta > 1/(n-1)$  an equilibrium with  $x^{\text{CS}^*} = 0$ ,  $x^{\text{OS}^*} > 0$  can not exist (see proposition 4.5). Finally if  $\theta > 1/(n-1)$ , then  $\beta(1+(n-1)\theta)/\eta < \beta 2\theta(n-1)/\eta$ . We can now conclude that if  $x_{\text{nc}}^{\text{OS}} < \beta(1+(n-1)\theta)/\eta$  an equilibrium with  $x^{\text{CS}^*} > 0$ ,  $x^{\text{OS}^*} = 0$  exists, as the two conditions of proposition 4.4 are fulfilled.  $\square$

**Proposition 4.8.** *If there is no code from the non-commercial community  $x_{\text{nc}}^{\text{OS}} = 0$ , then firms develop OS only if  $\theta < 1/(n+1)$ .*

*Proof.* With  $x_{\text{nc}}^{\text{OS}} = 0$  the upper conditions of both equilibria with OS, i.e.  $x^{\text{CS}^*} = 0$ ,  $x^{\text{OS}^*} > 0$  and  $x^{\text{CS}^*} > 0$ ,  $x^{\text{OS}^*} > 0$ , are fulfilled. With  $x_{\text{nc}}^{\text{OS}} = 0$  the lower boundary of both types of equilibria with OS are the same (compare 4.5 with 4.6) and yield the condition  $\beta(n-1)((n+1)\theta-1) < 0$ . This is fulfilled only if  $\theta < 1/(n+1)$ .  $\square$

The impact of  $x_{\text{nc}}^{\text{OS}}$  can be summarized as follows. For CS-only and OS-only Nash-equilibria (i.e. for  $x^{\text{CS}^*} > 0$ ,  $x^{\text{OS}^*} = 0$  and  $x^{\text{CS}^*} = 0$ ,  $x^{\text{OS}^*} > 0$ ) the following holds: Except for the case that OS-decisions are strategic complements ( $\phi < 2\phi_{\text{soc}}^{\text{OS}}$ ),  $x_{\text{nc}}^{\text{OS}}$  crowds out firm-developed code. In both cases firms substitute own developed code with the ‘cost free’-code  $x_{\text{nc}}^{\text{OS}}$ . In case of CS-OS equilibria ( $x^{\text{CS}^*} > 0$ ,  $x^{\text{OS}^*} > 0$ ) the impact of  $dx_{\text{nc}}^{\text{OS}} > 0$  is different: There is a crowding-out as well as a crowding-in, as  $\partial x^{\text{CS}^*}/\partial x_{\text{nc}}^{\text{OS}} > 0$  and  $\partial x^{\text{OS}^*}/\partial x_{\text{nc}}^{\text{OS}} < 0$ , with  $\partial x^{\text{CS}^*}/\partial x_{\text{nc}}^{\text{OS}} = -\partial x^{\text{OS}^*}/\partial x_{\text{nc}}^{\text{OS}}$ . Additionally, for  $1/(n+1) < \theta < 1/(n-1)$  an equilibrium with firm-developed OS code exists only if  $x_{\text{nc}}^{\text{OS}} > \beta((n+1)\theta-1)(n-1)/\eta$ . Otherwise the incentives to cooperate are too low and there is only competition on quality, hence CS software only. If  $\theta > 1/(n-1)$  there is no firm developed OS software at all, independent of  $x_{\text{nc}}^{\text{OS}}$ .

#### 4.4. OS and CS in Case of Restricted Licenses

In the case of restricted licenses we have to distinguish between CS-firms ( $i \in R$ ) and OS-firms ( $i \in Z$ ). Therefore we have to analyze (a) how much

software each type will develop in equilibrium, and (b) how many CS- and OS-firms will coexist. The latter refers to the question of the equilibrium proportion of OS-firms in the industry. We will examine this in section 5.2. In the current section we derive  $x^{\text{CS}*}$  and  $x^{\text{OS}*}$  for any given number of CS- and OS-firms.

We establish the equilibrium values for OS and CS in two steps. First we solve for optimal decision regarding the interaction among the CS-firms, and among the OS-firms:

- The symmetric Nash-equilibrium of CS (for a given amount of OS) is

$$x^{\text{CS}} = \frac{(1 + (n - 1)\theta)(\beta - z\theta X^{\text{OS}})}{\frac{1}{2}h^2\phi - (1 + (n - 1)\theta)(1 + z\theta)}$$

for all  $X^{\text{OS}} < \beta/z\theta$ , otherwise  $x^{\text{CS}} = 0$ .

- The symmetric Nash-equilibrium of OS (for a given amount of CS) is

$$x^{\text{OS}} = \frac{(1 + r\theta)(\beta - \theta \sum x^{\text{CS}}) - \left(\frac{1}{4}\phi h^2 - (1 + r\theta)^2\right) x_{\text{nc}}^{\text{OS}}}{\frac{1}{4}\phi h^2(1 + z) - z(1 + r\theta)^2}$$

for all  $\sum x^{\text{CS}} < \beta/\theta - \left(\frac{1}{4}\phi h^2 - (1 + r\theta)^2\right) x_{\text{nc}}^{\text{OS}} / (1 + r\theta)\theta$ , otherwise  $x^{\text{OS}} = 0$ .<sup>13</sup>

Because of symmetry we can replace  $X^{\text{OS}} = zx^{\text{OS}} + x_{\text{nc}}^{\text{OS}}$  and  $\sum x^{\text{CS}} = rx^{\text{CS}}$ . Furthermore, for convenience we replace the denominator of  $x^{\text{CS}}$  and  $x^{\text{OS}}$  with  $\psi$  and  $\chi$ . This yields

$$x^{\text{CS}} = \frac{(1 + (n - 1)\theta)(\beta - z\theta(zx^{\text{OS}} + x_{\text{nc}}^{\text{OS}}))}{\psi} \quad (12)$$

$$x^{\text{OS}} = \frac{(1 + r\theta)(\beta - \theta rx^{\text{CS}}) - \left(\frac{1}{4}\phi h^2 - (1 + r\theta)^2\right) x_{\text{nc}}^{\text{OS}}}{\chi} \quad (13)$$

As before, we will discuss the impact of quantity competition and quality competition on the OS and CS output. Assumed there are no CS firms

<sup>13</sup>As mentioned above we focus on cases where  $\phi > \phi_{\bar{x}}^{\text{OS}}$ . In the case of  $\phi < \phi_{\bar{x}}^{\text{OS}}$  multiple equilibria exist, see the Appendix.

( $r = 0$  and thus  $n = z$ ), then OS is again only affected by  $h$ , the degree of quantity competition, with  $\partial x^{os}/\partial h < 0$ . Only if the OS-firms face CS-competitors, they have to compete on quality (against the CS-firms). In case of CS, firms compete on quality regardless whether there are OS-firms or not. Hence, if  $z = 0$  we have  $\partial x^{cs}/\partial \theta > 0$  and  $\partial x^{cs}/\partial h < 0$ .

The next step takes into account the interaction between the CS- and OS-firms. It turns out that CS is a strategic substitute to OS and vice versa.

**Proposition 4.9.** *For CS development, OS code is a strategic substitute:  $\mathcal{E}_{os}^{cs} < 0$  and  $\mathcal{E}_{nc}^{cs} < 0$ .*

*Proof.* It is  $\mathcal{E}_{os}^{cs} = \partial x^{cs}/\partial x^{os} \cdot x^{cs}/x^{os} < 0$  and  $\mathcal{E}_{nc}^{cs} = \partial x^{cs}/\partial x_{nc}^{os} \cdot x_{nc}^{os}/x^{cs} < 0$ .  $\square$

**Proposition 4.10.** *For OS development, CS code is a strategic substitute:  $\mathcal{E}_{cs}^{os} < 0$ .*

*Proof.* It is  $\mathcal{E}_{cs}^{os} = \partial x^{os}/\partial x^{cs} \cdot x^{cs}/x^{os} < 0$  for all  $\chi > 0$ . (For  $\chi < 0$ , i.e. the case where OS firms produce the cutoff, see the Appendix.)  $\square$

Now, (12) and (13) yield the Nash-equilibria in the simultaneous decision about OS and CS for a given number of OS- and CS-firms. There exist equilibria where only one type of firm develops software as well as where both types develop code. In particular:

**Proposition 4.11.** *A Nash-equilibrium where only the CS-firms develop software ( $x^{cs*} > 0$ ,  $x^{os*} = 0$ ) exists in the case that the*

- *OS-decisions are strategic complements or relatively weak strategic substitutes ( $\mathcal{E}_{nc}^{os} > -\mathcal{E}_{nc}^{cs}\mathcal{E}_{cs}^{os}$ ), if and only if  $x_{nc}^{os} < \beta/z\theta$  and also  $x_{nc}^{os} < \beta(1+r\theta)\kappa/\mu$ ,*
- *OS-decisions are relatively strong strategic substitutes ( $\mathcal{E}_{nc}^{os} < -\mathcal{E}_{nc}^{cs}\mathcal{E}_{cs}^{os}$ ), if and only if  $\beta(1+r\theta)\kappa/\mu < x_{nc}^{os} < \beta/z\theta$ ,*

with  $\kappa = \frac{1}{2}h^2\phi - (1+(n-1)\theta)(1+\theta n)$ , and  $\mu = \frac{1}{2}h^2\phi[\frac{1}{4}h^2\phi - (1+r\theta)^2] - (1+(n-1)\theta)[\frac{1}{4}h^2\phi(1+z\theta) - (1+r\theta)(1+\theta n)]$ .

*Proof.* If only the CS-firms develop software (thus  $x^{os*} = 0$ ) then  $x^{cs*}$  is given by

$$x^{cs*} = \frac{(1+(n-1)\theta)(\beta - z\theta x_{nc}^{os})}{\frac{1}{2}h^2\phi - (1+(n-1)\theta)(1+z\theta)}$$



First,  $x^{\text{CS}*}$  has to be greater zero, which is fulfilled if  $x_{\text{nc}}^{\text{OS}} < \beta/z\theta$ . Second, the corresponding  $x^{\text{OS}*}$  must be zero, from (13) we obtain  $r \cdot x^{\text{CS}*} > \beta/\theta - \left(\frac{1}{4}\phi h^2 - (1+r\theta)^2\right) x_{\text{nc}}^{\text{OS}} / (1+r\theta)\theta$ . Inserting the above expression of  $x^{\text{CS}*}$  and solving for  $x_{\text{nc}}^{\text{OS}}$  leads to  $x_{\text{nc}}^{\text{OS}} < \beta\kappa/\mu$  if  $\mu > 0$ , and  $x_{\text{nc}}^{\text{OS}} > \beta\kappa/\mu$  otherwise. Now,  $\mu > 0$  if  $(1+(n-1)\theta)z\theta / \left[\frac{1}{2}h^2\phi - (1+(n-1)\theta)(1+z\theta)\right] > \left(\frac{1}{4}\phi h^2 - (1+r\theta)^2\right) / (1+r\theta)r\theta$ , which can be rewritten as  $\mathcal{E}_{\text{nc}}^{\text{OS}} > -\mathcal{E}_{\text{nc}}^{\text{CS}}\mathcal{E}_{\text{cs}}^{\text{OS}}$ .  $\square$

The interpretation of proposition 4.11 is as follows. *First*, there must be CS-development. Recall that  $\mathcal{E}_{\text{nc}}^{\text{CS}} < 0$ , i.e. OS software from the non-commercial community has a negativ impact on CS output. Thus, it must be that  $x_{\text{nc}}^{\text{OS}} < \beta/z\theta$ , otherwise  $x_{\text{nc}}^{\text{OS}}$  suppresses code-development by CS-firms, regardless of additional OS code by firms. *Second*, the developed CS code must be enough to suppress OS-development by firms. Suppose that  $x_{\text{nc}}^{\text{OS}} = 0$ . In such a case  $x^{\text{CS}*}$  supresses firm-OSS if  $rx^{\text{CS}*} > \beta/\theta$ . This condition is fulfilled if the marginal costs of software-development rise relatively slowly, namely if  $\phi < \phi_{\text{soc}}^{\text{CS}} \cdot (1+n\theta)/(1+(n-1)\theta)$ . If the marginal costs rise faster, then the cost-sharing aspect of OS becomes so dominant that the OS-firms always develop some code. In case of  $x_{\text{nc}}^{\text{OS}} > 0$  one has to take into account that  $x_{\text{nc}}^{\text{OS}}$  has an direct and an indirect impact on  $x^{\text{OS}}$ . The indirect effect is that  $x_{\text{nc}}^{\text{OS}}$  affects  $x^{\text{CS}}$  and  $x^{\text{CS}}$  in turn affects  $x^{\text{OS}}$ . Via this indirect effect  $x_{\text{nc}}^{\text{OS}}$  has a positive impact on  $x^{\text{OS}}$ , because  $x_{\text{nc}}^{\text{OS}}$  decreases  $x^{\text{CS}}$  ( $\mathcal{E}_{\text{nc}}^{\text{CS}} < 0$ ), and this decrease has a positive impact on  $x^{\text{OS}}$  ( $\mathcal{E}_{\text{cs}}^{\text{OS}} < 0$ ). The direct effect of  $x_{\text{nc}}^{\text{OS}}$  on  $x^{\text{OS}}$  is expressed by  $\mathcal{E}_{\text{nc}}^{\text{OS}}$  and can be positive or negative. If OS-decisions are only weak strategic substitutes or even strategic complements, then the overall effect is positive. Thus, to ensure that  $x^{\text{OS}} = 0$  there must be not too much  $x_{\text{nc}}^{\text{OS}}$ . But if  $x_{\text{nc}}^{\text{OS}}$  strongly crowds out firm-OSS ( $\mathcal{E}_{\text{nc}}^{\text{OS}} < -\mathcal{E}_{\text{nc}}^{\text{CS}}\mathcal{E}_{\text{cs}}^{\text{OS}}$ ) then  $x_{\text{nc}}^{\text{OS}}$  helps to suppress firm-OS. (If OS-decisions are very strong strategic substitutes, then firm-OS is suppressed even for  $x_{\text{nc}}^{\text{OS}} = 0$ . Formally:  $\beta(1+r\theta)\kappa/\mu < 0$ ).

**Proposition 4.12.** *A Nash-equilibrium where only the OS-firms develop software ( $x^{\text{CS}*} = 0, x^{\text{OS}*} > 0$ ) exists in the case that*

- OS-decisions are strategic substitutes ( $\mathcal{E}_{\text{nc}}^{\text{OS}} < 0$ ), if and only if  $\beta\sigma/\frac{1}{4}h^2\phi\theta < x_{\text{nc}}^{\text{OS}} < \beta(1+r\theta)/\left(\frac{1}{4}h^2\phi - (1+r\theta)^2\right)$ .
- OS-decisions are strategic complements ( $\mathcal{E}_{\text{nc}}^{\text{OS}} > 0$ ), if and only if  $\beta\sigma/\frac{1}{4}h^2\phi\theta < x_{\text{nc}}^{\text{OS}}$ .

with  $\sigma = \chi - z^2\theta(1+r\theta)$

*Proof.* If only the OS-firms develop software, hence  $x^{cs*} = 0$ , then  $x^{os*}$  is given by

$$x^{os*} = \frac{(1+r\theta)\beta - \left(\frac{1}{4}\phi h^2 - (1+r\theta)^2\right)x_{nc}^{os}}{\chi}.$$

First,  $x^{os*}$  has to be greater zero, which implies  $x_{nc}^{os} < (1+r\theta)\beta / \left(\frac{1}{4}h^2\phi - (1+r\theta)^2\right)$  if OS-decisions are strategic substitutes.<sup>14</sup> If they are strategic complements, then  $x^{os*} > 0 \forall x_{nc}^{os}$ . Second, the corresponding  $x^{cs*}$  must be zero. We obtain from (12) the condition  $zx^{os*} + x_{nc}^{os} > \beta/z\theta$ , and this finally yields  $x_{nc}^{os} > (\chi - z^2\theta(1+r\theta))\beta / \frac{1}{4}h^2\phi\theta$ .  $\square$

Proposition 4.12 is straightforward. If OS-decisions are strategic complements, then any  $x_{nc}^{os}$  fosters OS-development by firms. If they are strategic substitutes,  $x_{nc}^{os}$  crowds out firm-OS. To ensure that the CS-firms do not develop any code, total OS ( $X^{os}$ ) must be greater than  $\beta/z\theta$ . As a consequence there is the need for sufficient community-code, if the joint code output of the OS-firms is not enough. There is no need for community-code to fulfill the condition if the marginal costs of software-development rise relatively slowly: for  $\phi < (1+n\theta)/(1+r\theta) \cdot z/(z+1) \cdot 2 \cdot \phi_{soc}^{os}$  OS-firms jointly produce enough code to suppress CS. In this case is  $(\chi - z^2\theta(1+r\theta))\beta / \frac{1}{4}h^2\phi\theta < 0$  and the condition is therefore fulfilled  $\forall x_{nc}^{os}$ .

**Proposition 4.13.** *A Nash-equilibrium where both types of firm develop software ( $x^{cs*} > 0, x^{os*} > 0$ ) exists*

(a) for  $\mathcal{E}_{cs}^{os} \mathcal{E}_{os}^{cs} < 1$  in the case that

- OS-decisions are strategic complements or relatively weak strategic substitutes ( $\mathcal{E}_{nc}^{os} > -\mathcal{E}_{nc}^{cs} \mathcal{E}_{cs}^{os}$ ), if and only if  $\beta(1+r\theta)\kappa/\mu < x_{nc}^{os} < \beta\sigma/\frac{1}{4}h^2\phi\theta$ ,
- OS-decisions are relatively strong strategic substitutes ( $\mathcal{E}_{nc}^{os} < -\mathcal{E}_{nc}^{cs} \mathcal{E}_{cs}^{os}$ ), if and only if  $x_{nc}^{os} < \beta\sigma/\frac{1}{4}h^2\phi\theta$  and also  $x_{nc}^{os} < \beta(1+r\theta)\kappa/\mu$ ,

(b) for  $\mathcal{E}_{cs}^{os} \mathcal{E}_{os}^{cs} > 1$  in the case that

- OS-decisions are strategic complements or relatively weak strategic substitutes ( $\mathcal{E}_{nc}^{os} > -\mathcal{E}_{nc}^{cs} \mathcal{E}_{cs}^{os}$ ), if and only if  $\beta\sigma/\frac{1}{4}h^2\phi\theta < x_{nc}^{os} < \beta(1+r\theta)\kappa/\mu$ ,

<sup>14</sup>If  $x_{nc}^{os}$  exceeds this level, then there is so much community-code that the OS-firms do not have any incentives to develop own code.

- OS-decisions are relatively strong strategic substitutes ( $\mathcal{E}_{nc}^{os} < -\mathcal{E}_{nc}^{cs}\mathcal{E}_{cs}^{os}$ ), if and only if  $x_{nc}^{os} > \beta\sigma/\frac{1}{4}h^2\phi\theta$  and also  $x_{nc}^{os} > \beta(1+r\theta)\kappa/\mu$ ,

with  $\sigma = \chi - z^2\theta(1+r\theta)$ , and  $\kappa = \frac{1}{2}h^2\phi - (1+(n-1)\theta)(1+\theta n)$ , and  $\mu = \frac{1}{2}h^2\phi[\frac{1}{4}h^2\phi - (1+r\theta)^2] - (1+(n-1)\theta)[\frac{1}{4}h^2\phi(1+z\theta) - (1+r\theta)(1+\theta n)]$ .

*Proof.* The simultaneous solution of 12 and 13 is given by

$$x^{cs*} = (1+(n-1)\theta) \frac{\beta\sigma - z\theta\frac{1}{4}\phi h^2 x_{nc}^{os}}{\psi\chi - (1+\theta r)\theta^2 r(1+(n-1)\theta)z^2}, \quad (14)$$

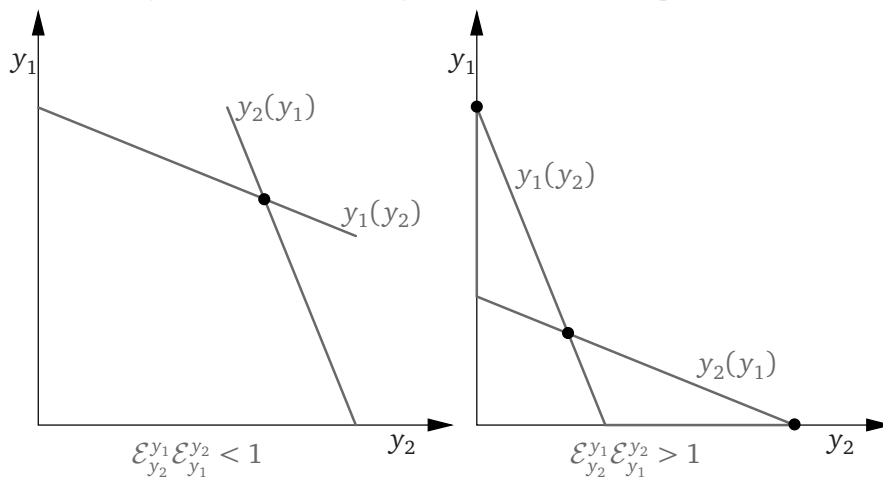
$$x^{os*} = \frac{\kappa\beta(1+\theta r) - \mu x_{nc}^{os}}{\psi\chi - (1+\theta r)\theta^2 r z^2(1+(n-1)\theta)}. \quad (15)$$

The denominator of both expressions is greater zero if and only if  $\mathcal{E}_{cs}^{os}\mathcal{E}_{os}^{cs} > 1$ .<sup>15</sup> Case  $\mathcal{E}_{cs}^{os}\mathcal{E}_{os}^{cs} > 1$ : If the denominator is positive, then  $x^{cs}$  and  $x^{os}$  are positive if also the numerators have a positive sign.  $x^{cs*} > 0$  leads to the condition  $x_{nc}^{os} < \beta\sigma/\frac{1}{4}h^2\phi\theta$ . The condition for  $x^{os*} > 0$  depends on whether  $\mu > 0$  or not. This is greater zero if  $\mathcal{E}_{nc}^{os} < -\mathcal{E}_{nc}^{cs}\mathcal{E}_{cs}^{os}$  which then implies that  $x_{nc}^{os} < \beta(1+r\theta)\kappa/\mu$  is the condition for  $x^{os*} > 0$ . Otherwise the opposite holds. Case  $\mathcal{E}_{cs}^{os}\mathcal{E}_{os}^{cs} < 1$ : If the denominator is negative, then  $x^{cs}$  and  $x^{os}$  are positive if also the numerators have a negative sign. Thus the unequal signs of the conditions are reversed.  $\square$

The logic behind proposition 4.13 is the following. OS and CS are linear functions of each other, and OS and CS react on each other as strategic substitutes (see propositions 4.9 and 4.10). A Nash-equilibrium with  $(x^{cs*} > 0, x^{os*} > 0)$  can thus exist only if either (a) neither  $(x^{cs*} > 0, x^{os*} = 0)$  nor  $(x^{cs*} = 0, x^{os*} > 0)$  exist, or (b) both  $(x^{cs*} > 0, x^{os*} = 0)$  and  $(x^{cs*} = 0, x^{os*} > 0)$  exist. Figure 2 illustrates the underlying logic with an symmetric example of decisions in strategic substitutes. In the left hand side of figure 2 there is only one equilibrium: the inner solution  $y_1^* > 0, y_2^* > 0$ . In the right hand side there are three equilibria. The inner solution  $(y_1^* > 0, y_2^* > 0)$  as well as the two corner solutions  $y_1^* > 0, y_2^* = 0$  and  $y_1^* = 0, y_2^* > 0$ . Applying this to our model implies that a Nash-equilibrium where both types of firms develop code exists if (a) neither

<sup>15</sup>Notice that we have defined the elasticities with respect to (12) and (13). Thus the symmetry of OS-firms, and of CS-firms, each was taken into account. Namely the derivatives are  $\partial x^{cs}/\partial x^{os} = -(1+(n-1)\theta)z^2\theta/\psi$  and  $\partial x^{os}/\partial x^{cs} = -(1+r\theta)r\theta/\psi$ .

Figure 2: Linear Strategic Substitutes and Equilibrium



proposition 4.11 nor 4.12 are fulfilled, or (b) proposition 4.11 and 4.12 are fulfilled simultaneously. From this, including the non negativity conditions, one obtains the conditions of the above proposition.

This section has analyzed the equilibria in code production for restricted licenses. It turns out that such licenses stabilize firm contributions to OS: If there are only OS-firms ( $r = 0$ ), then firms develop OS code unless there is enough community code. Thus if  $x_{nc}^{OS} = 0$ , hence if no non-commercial community exists, the OS-firms always produce OS, see (13). Also if OS-firms compete with CS-firms there can be firm-OS if  $x_{nc}^{OS} = 0$ . However, the impact of community-code on the different equilibria is more complex as with liberal licenses. The reason for this is that  $x_{nc}^{OS}$  directly affects optimal OS, via this indirectly CS, on which in turn OS again reacts.

## 5. Mixed Industries

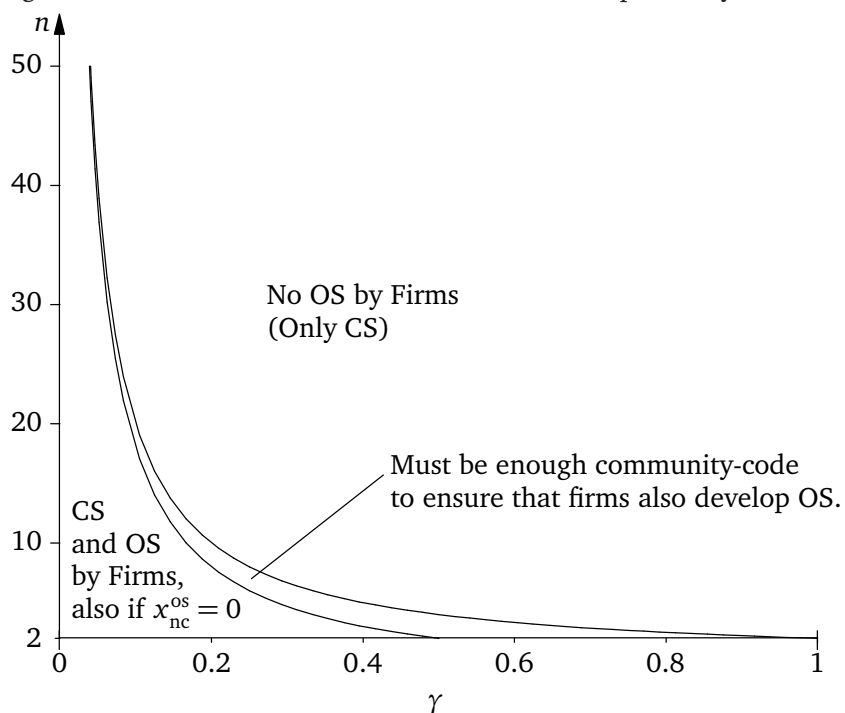
This section analyzes mixed industries. Throughout the paper a mixed industry is defined as an industry with  $n$  firms and OS as well as CS production by firms in equilibrium. Of course, we have to distinguish whether the OS license is liberal or restricted:

### 5.1. Liberal Licenses: OS and CS Development

If licenses are liberal, then all firms can develop OS as well as CS code. Recall, that here the degree of quality competition ( $\theta$ ) has a strongly negative impact on firm-OS. For all  $\theta > 1/(n-1)$  firms develop only CS code (proposition 4.7). Furthermore, if there is no community-code ( $x_{nc}^{os} = 0$ ) firms also develop no OS code unless  $\theta < 1/(n+1)$ . (proposition 4.8). This implies that for  $1/(n+1) < \theta < 1/(n-1)$  a mixed industry exists only if  $x_{nc}^{os}$  has the proper value, while only for  $\theta < 1/(n+1)$  a mixed industry exists also in the absence of a non-commercial oriented community.

As  $\theta = \gamma/(2-\gamma)$ , the above conditions are equivalent to  $\gamma < 2/(n+2)$  and  $\gamma < 2/n$ . Figure 3 gives a graphical impression of the two conditions. Unless industries are very concentrated (small  $n$ ), markets are must be very

Figure 3: Liberal Licenses: OS and CS Code Development by Firms



separated (high horizontal product differentiation): for a  $n = 10$  industry the thresholds for the (horizontal) degree of substitution are  $\gamma = 0.16\bar{6}$  and  $\gamma = 0.2$ , while for a  $n = 100$  industry the thresholds are  $\gamma = 0.0196$  and  $\gamma = 0.02$ . Finally,  $\lim_{n \rightarrow \infty} 2/(n+2) = 0$  and  $\lim_{n \rightarrow \infty} 2/(n) = 0$ .

## 5.2. Restricted Licenses: Equilibrium of OS- and CS-Firms

This subsection analyzes the ratio of OS- and CS-firms, assumed free entry and exit. We will not be explicit about the entry process as such and thus also ignore possible historical events (lock-ins) in this paper.<sup>16</sup> We analyze the condition for a stable mixed industry, i.e. a situation that resists further entry by OS- and CS-firms. An  $n$ -firm industry with  $z$  OS-firms and  $r$  CS-firms is an equilibrium if the incumbents earn profits  $\pi_i \geq 0$  and an additional OS- or CS-entrant would earn negative profits. This condition is fulfilled if  $\pi_i = 0 \forall i \in N$ . This is a sufficient condition, and it is also the necessary condition if  $n$  is large.<sup>17</sup> Therefore, this paper concentrates on the zero-profit condition<sup>18</sup>

$$\pi_{i \in Z} = p_{i \in Z} \cdot q_{i \in Z} - c_{i \in Z} - C = \pi_{i \in R} = p_{i \in R} \cdot q_{i \in R} - c_{i \in R} - C = 0. \quad (16)$$

Furthermore, we concentrate on Nash-equilibria where both types of firm develop code:  $(x^{cs*} > 0, x^{os*} > 0)$ . The reason is the following: If there is no community-code ( $x_{nc}^{os} = 0$ ), then  $(x^{cs*} > 0, x^{os*} = 0)$  and  $(x^{cs*} = 0, x^{os*} > 0)$  violate the zero-profit condition. The following lemmas 5.1 and 5.2 show this, with  $\beta$  normalized to  $\beta = 1$  without loss of generality.

**Lemma 5.1.** *Given a Nash-equilibrium where only the CS-firms develop software ( $x^{cs*} > 0, x^{os*} = 0$ ), the CS-firms earn higher profits than the OS-firms for  $\beta = 1$  and  $x_{nc}^{os} = 0$ .*

<sup>16</sup>The question of possible lock-ins, strategic OS-versus-CS decision of incumbents etc. is analyzed in MAURER.

<sup>17</sup> For small  $n$  the sufficient and necessary condition is:  $\pi_i \geq 0 \forall i \in N$ ,  $\pi_e < 0$  with  $e \notin N$  is either a CS- or an OS-entrant. For  $n \rightarrow \infty$  the sufficient and necessary conditions converge to  $\pi_i = 0$ .

<sup>18</sup>A discussion of  $\pi_i > 0 \forall i \in N$ ,  $\pi_e < 0$  with  $e \notin N$  is either a CS- or an OS-entrant can be found in MAURER.

*Proof by contradiction.* With  $\beta = 1$  and  $x_{nc}^{os} = 0$  profits for  $x^{cs} > 0$ ,  $x^{os} = 0$  are  $\pi_{i \in Z} = (1-r\theta x^{cs})^2/h^2$  and  $\pi_{i \in R} = (1+x^{cs}(1+z\theta))^2/h^2 - x^{cs2}\phi/2$ . Now let us assume that  $\pi_{i \in Z} > \pi_{i \in R}$ . There are two necessary conditions for this:

- (a) OSS-firms achieve positive prices. This is true if and only if  $x^{cs} < 1/r\theta$ .
- (b)  $\pi_{i \in Z} > \pi_{i \in R}$  implies that  $x^{cs} > 4(1+n\theta)/[\phi h^2 + 2(r^2\theta^2 - (1+z\theta)^2)]$ .

The conditions (a) and (b) are simultaneously fulfilled if and only if  $\phi > 2(1+n\theta)^2/h^2$ . With  $\beta = 1$  and  $x_{nc}^{os} = 0$  proposition 4.11 leads to the condition  $\phi < 2(1+n\theta)(1+(n-1)\theta)/h^2$ . But  $2(1+n\theta)^2/h^2 > 2(1+n\theta)(1+(n-1)\theta)/h^2$ . We can now conclude that (a) and (b) are simultaneously fulfilled, i.e. that  $\pi_{i \in Z} > \pi_{i \in R}$ , if and only if a Nash-equilibrium with  $x^{cs*} > 0$ ,  $x^{os*} = 0$  does not exist.  $\square$

**Lemma 5.2.** *Given a Nash-equilibrium where only the OS-firms develop software ( $x^{cs*} = 0$ ,  $x^{os*} > 0$ ), the OS-firms earn higher profits than the CS-firms for  $\beta = 1$  and  $x_{nc}^{os} = 0$ .*

*Proof by contradiction.* With  $\beta = 1$  and  $x_{nc}^{os} = 0$  profits for  $x^{cs} = 0$ ,  $x^{os} > 0$  are  $\pi_{i \in Z} = (1+(1+r\theta)zx^{os})^2/h^2 - 1/2\phi zx^{os2}$  and  $\pi_{i \in R} = (1-z^2\theta x^{os})^2/h^2$ . Now let us assume that  $\pi_{i \in Z} < \pi_{i \in R}$ . There are two necessary conditions for this:

- (a) CSS-firms achieve positive prices. This is true if and only if the condition  $x^{os} < 1/(z^2\theta)$  is fulfilled.
- (b)  $\pi_{i \in Z} < \pi_{i \in R}$  implies that  $x^{os} > 4(1+n\theta)/[\phi h^2 + 2z(z^2\theta^2 - (1+r\theta)^2)]$  holds.

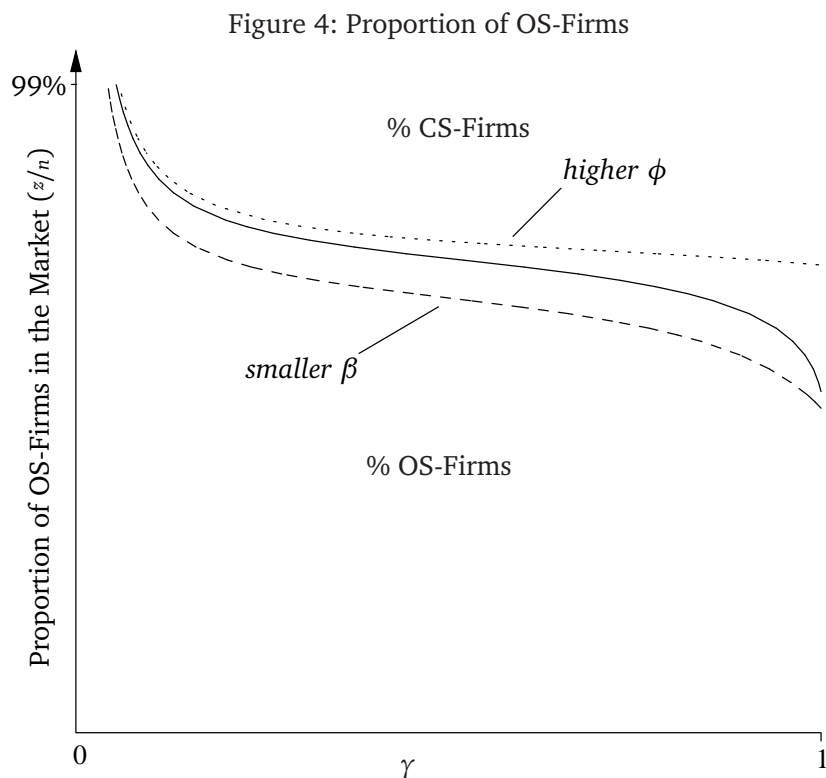
The conditions (a) and (b) are simultaneously fulfilled if and only if  $\phi > 2z(1+n\theta)^2/h^2$ . With  $\beta = 1$  and  $x_{nc}^{os} = 0$  proposition 4.12 yields the condition  $\phi < 2z/(1+z) \cdot 2(1+n\theta)(1+r\theta)/h^2$ . But  $2z(1+n\theta)^2/h^2 > 2z/(1+z) \cdot 2(1+n\theta)(1+r\theta)/h^2$ . We can now conclude that (a) and (b) are simultaneously fulfilled, i.e. that  $\pi_{i \in Z} < \pi_{i \in R}$ , if and only if a Nash-equilibrium with  $x^{cs*} = 0$ ,  $x^{os*} > 0$  does not exist.  $\square$

Furthermore, if there is community code, then obviously lemma 5.2 still holds. Lemma 5.2 has to be modified only slightly: If there is a sufficient high amount of community-OS, then the OS-firms can survive in the market without contributing. Allegorically spoken, OS-firms in such a case live in Cockaigne (land of plenty) as they get enough code “for free”. However, as this is a trivial case we exclude this from the following analysis.

The zero-profit condition (16) leads to

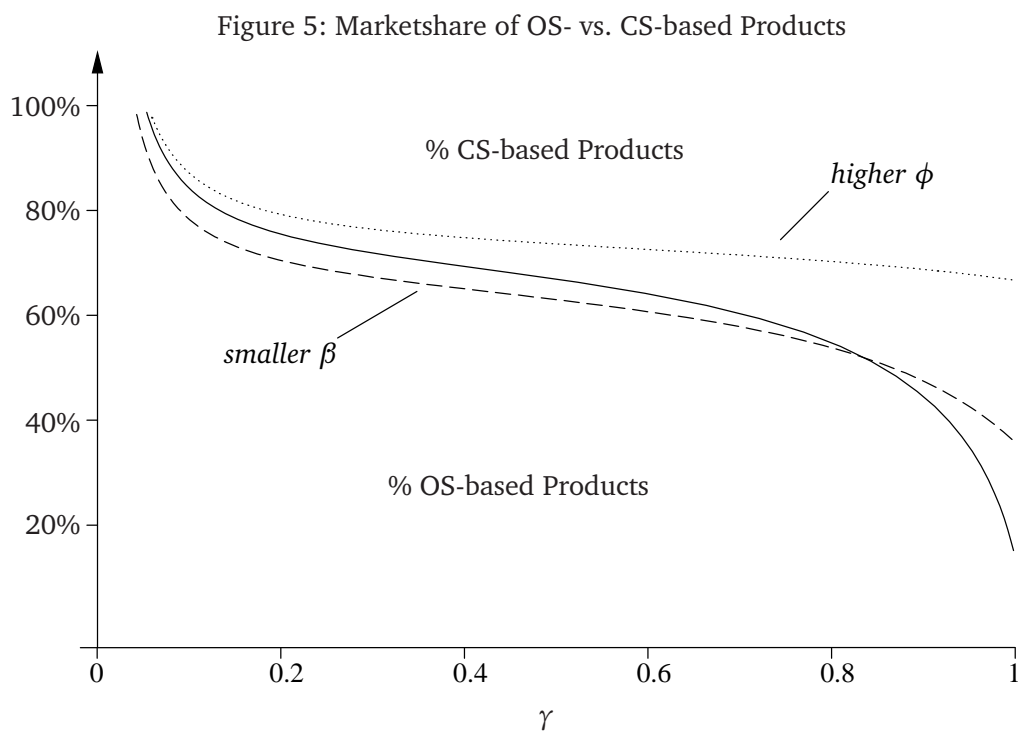
$$P_{i \in Z} \cdot q_{i \in Z} - c_{i \in Z} = P_{i \in R} \cdot q_{i \in R} - c_{i \in R}.$$

We use this to numerically calculate stable mixed industries. Figure 4 depicts the typical example of an industry with  $n = 100$  firms. The solid line



is the outcome for  $\phi = 2$ ,  $\beta = 1$ , and  $x_{nc}^{os} = 0$ . ( $x_{nc}^{os}$  was set equal zero to ensure that for all parameters no ‘Cockaigne-situation’ for the OS-firms exist. The impact of spooky is explained below.) As the reader can confirm by inspection, the proportion of CS-firms decreases when the products produced in stage two become closer substitutes. The corresponding figure 5 shows the total market share of products (bundles) with OS or CS code. For industries with high  $\gamma$ , thus with only a low degree of horizontal





product differentiation, the market share of CS-based products exceeds the OS-based ones. For  $\gamma$  close to one, 80% or more of the products are based on CS, while less than half of the firms in the market are CS-firms. In other words: some ‘big’ CS-firms compete with ‘small’ OS-firms. Furthermore, CS-firms offer higher quality (more code per bundle). The latter statement still holds when the parameters  $\phi$  and  $\beta$  are changed, while the proportion of OS-firms in the industry and the market share of OS-based products differ. Figures 4 and 5 also depict the impact of different  $\phi$  and  $\beta$ :

The dotted line represents the case of  $\phi = 5$ , i.e. when the marginal costs of software development increase more steeply. As result, the number of OS-firms in the industry as well as the market share of OS-based products is higher. The reason for this is that with higher  $\phi$  quality-investments (software-development) become more costly (costs increase more steeply)

and thus the cost-sharing and quality-cooperation aspect of OS becomes more attractive. Therefore more OS-firms are in equilibrium in the industry, with each firm's market share as well as the total market share of OS-based products being higher.

The dashed line represents a lower quality of the stage two product:  $\beta = 0.8$ . Recall, that the quality of the bundles ( $\alpha_i = \beta + x_i$ ) directly affects demand:  $p_i = \alpha_i - q_i - \gamma \sum_{j \neq i} q_j$ . Thus, a lower  $\beta$  means that the complementary private good is of less importance for generating revenues. Hence, the software-decision pays more. As one would expect, this decreases the number of firms running OS-business models. On the other hand, when software-development becomes more important, then the remaining OS-firms have a higher incentive to develop code as they compete on quality with the CS-firms. This explains why for higher  $\gamma$  the market share of OS-based products is higher compared to the  $\beta = 1$  case. Here the OS-firms together produce more code (offer higher quality) as they would do in the  $\beta = 1$  case.

Finally the impact of community-code can be analyzed with numerical examples. The effect is straightforward:  $x_{nc}^{os} > 0$  makes OS-business models more attractive. As result, the number of OS-firms in the industry as well as their total and individual market share increase. Furthermore, even for small  $x_{nc}^{os}$ , firms stop developing OS-code if the products becomes more substitutes. In these cases OS-firms survive in the market without the need to invest. In such cases OS-firms do not contribute to the OS software but just use it.

## 6. Summary and Outlook

The paper analyzes the *economics of open versus closed source business models* with a general model. In stage one firms develop software, either as OS or CS code, or as a mix of OS and CS code. In stage two firms bundle this 'stage one'-software with complementary products (hardware, service, or proprietary software), and then compete. Competition in stage two is modeled as oligopolistic competition. We allow for *horizontal* product differentiation. Furthermore, firm  $i$ 's software developed in stage one affect quality and hence increase consumers' willingness to pay for  $q_i$  and yields

*vertical* product differentiation. The results are the following:

- The rationale of OS business models can be explained with a general two-stage model that combines aspects of non-cooperative R&D with the theory of differentiated oligopolies. OS enables firms to cooperate on quality, hence to avoid quality competition.
- If licenses allow a direct mix of OS with CS code, then there is OS development by firms only if the degree of quality competition is low. Otherwise a public good dilemma occurs: firms use given OS-code if there is one, but produce only CS code. Restricted licenses ensure OS-development.
- A firm's CS code is always a strategic substitute to other firms' CS code, whereas a firm's OS can be a strategic complement to other firms' OS code.
- Assumed free entry and exist, and that the OS-license is restricted, then equilibria of mixed industries (OS- and CS-firms) exist. Here OS-firms offer lower quality than their CS-rivals. If the products are close substitutes then CS-based products have the major market share.

The paper does not calculate welfare. However, the fact that in industry equilibrium the OS-firms are the low-quality providers offers a first hint towards the result: Compared to social optimum, there are too many OS-firms in the market. Furthermore, the paper is also not explicit about the entry process and thus ignore possible historical events like lock-ins. Also strategic OS- versus CS-decisions by incumbents are not taken into account. Thus, it is possible that industries stick in pure rather than mixed equilibria. These aspects are analyzed in [von Engelhardt and Maurer \(2010\)](#).

Finally, the paper analyzes the incentives for firms doing OS based on simple economic incentives only. Thus, social interactions and community-norms are not taken into account. It is well known, that the OS-community pays attention to what e.g. firms do and at least some kind of contribution may be expected by the community. Thus, social norms can play a role here, as breaking the rules will be sanctioned by the community, that is stop cooperating or migrate to other projects ([Osterloh et al., 2001](#), p 16

f). In other words: Firms might contribute OS code in more cases of liberal licenses than the model predicts. However, it is still true that the simple Cournot logic has surprisingly high explanatory regarding the question why firms contribute to OS. Also for cases where no non-commercial community delivers some code for free, the model explains the rationale of OS-based business models in a general framework. And finally the model points to the importance of formal institutions, namely the type of OS license.

## References

- Allarakhia, M., Kilgour, D.M., Fuller, J.D., 2010. Modelling the incentive to participate in open source biopharmaceutical innovation. *R&D Management* 40, 50 – 66.
- Baake, P., Wichmann, T., 2004. *Open Source Software, Competition and Potential Entry*. Berlecon Research Papers 5. Berlecon Research. Berlin.
- Berends, P., van Wegberg, M., 2000. *Competing Communities of Users and Developers of Computer Software. Competition Between Open Source Software and Commercial Software*. NIBOR Working Paper 1. Netherlands Institute of Business Organization and Strategy Research. Maastricht.
- Bertacchini, E.E., 2008. Coase, pigou and the potato: Whither farmers' rights? *Ecological Economics* 68, 183 – 193.
- Bessen, J., 2006. Open source software: Free provision of complex public goods, in: Bitzer, J., Schröder, P. (Eds.), *The Economics Of Open Source Software Development*. Elsevier, p. 57–81.
- Bitzer, J., 2004. Commercial versus open source software: the role of product heterogeneity in competition. *Economic Systems* 28, 369–381.
- de Bondt, R., 1997. Spillovers and innovative activities. *International Journal of Industrial Organization* 15, 1–28.
- Brand, A., Schmid, A., 2005. *Koordination in Einem Open Source-Projekt*. Technical Report.
- Bulow, J.I., Geanakoplos, J.D., Klemperer, P.D., 1985. Multimarket oligopoly: Strategic substitutes and complements. *The Journal of Political Economy* 93, 488–511.
- Casadesus-Masanell, R., Ghemawat, P., 2006. Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows. *Management Science* 52, 1072–1084.

- Casadesus-Masanell, R., Llanes, G., 2009. Mixed Source. Working Papers 09-06. NET Institute.
- Dahlander, L., Wallin, M.W., 2006. A man on the inside: Unlocking communities as complementary assets. *Research Policy* 35, 1243 – 1259.
- d'Aspremont, C., Jacquemin, A., 1988. Cooperative and noncooperative r&d in duopoly with spillovers. *American Economic Review* 78, 1133–37.
- David, P.A., Shapiro, J.S., 2008. Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy* 20, 364–398.
- Deroian, F., Gannon, F., 2006. Quality-improving alliances in differentiated oligopoly. *International Journal of Industrial Organization* 24, 629–637.
- Dixit, A., 1979. A model of duopoly suggesting a theory of entry barriers. *Bell Journal of Economics* 10, 20–32.
- Economides, N., Katsamakas, E., 2006. Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science* 52, 1057 – 1071.
- von Engelhardt, S., 2008a. The Economic Properties of Software. Jena Economic Research Papers.
- von Engelhardt, S., 2008b. Intellectual Property Rights and Ex-Post Transaction Costs: the Case of Open and Closed Source Software. Jena Economic Research Papers 2008-047. Friedrich-Schiller-University Jena, Max-Planck-Institute of Economics.
- von Engelhardt, S., Freytag, A., 2010. Institutions, Culture, and Open Source. Jena Economic Research Papers 2010-010. Friedrich-Schiller-University Jena and Max-Planck-Institute of Economics.
- von Engelhardt, S., Maurer, S.M., 2010. The New (Commercial) Open Source: Does it Really Improve Social Welfare? Berkeley Goldman School of Public Policy Working Paper 10-001.

- Fosfuri, A., Giarratana, M.S., Luzzi, A., 2005. Firm Assets and Investments in Open Source Software Product. DRUID Working Papers 05-10.
- Fosfuri, A., Giarratana, M.S., Luzzi, A., 2008. The penguin has entered the building: The commercialization of open source software products. *Organization Science* 19, 292–305.
- Gehring, R.A., 2006. The institutionalization of open source. *Poiesis & Praxis: International Journal of Technology Assessment and Ethics of Science* 4, 54–73.
- Ghosh, R.A., Glott, R., Krieger, B., Robles, G., 2002. FLOSS Final Report, Part 4: Survey of developers. Technical Report. International Institute of Infonomics, University of Maastricht.
- Häckner, J., 2000. A note on price and quantity competition in differentiated oligopolies. *Journal of Economic Theory* 93, 233–239.
- Harison, E., Cowan, R., 2004. On substitution of intellectual property and free disclosure: An analysis of R&D strategies in software technologies. *Economics of Innovation and New Technology* 13, 477 – 487.
- Harison, E., Koski, H., 2010. Applying open innovation in business strategies: Evidence from finnish software firms. *Research Policy* 39, 351 – 359.
- Henkel, J., 2006. The Jukebox Mode of Innovation - a Model of Commercial Open Source Development. DRUID Working Papers 06-25. DRUID, Copenhagen Business School, Department of Industrial Economics and Strategy/Aalborg University, Department of Business Studies.
- Henkel, J., Maurer, S.M., 2007. The Economics of Synthetic Biology. Technical Report Article number: 117.
- von Hippel, E., 2005. Open source software projects as user innovation networks - no manufacturer required, in: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (Eds.), *Perspectives On Free and Open Source Software*. MIT Press, Cambridge, Mass. [u.a.].

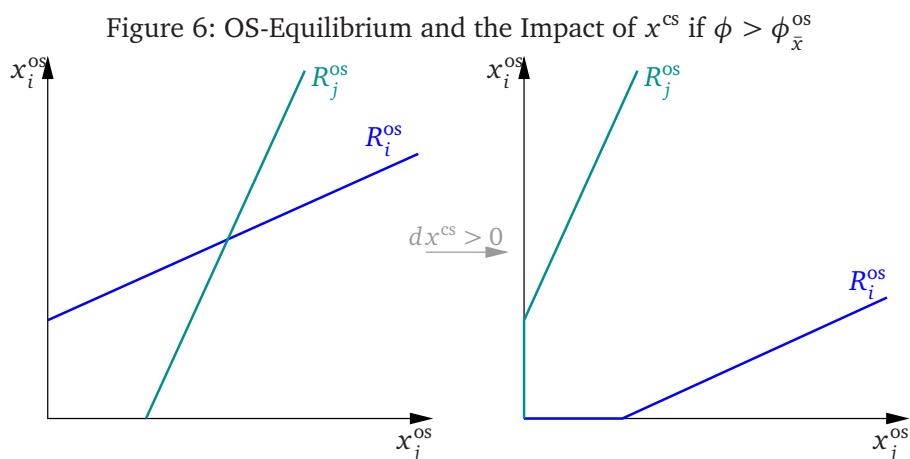
- von Hippel, E., von Krogh, G., 2003. Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science* 14, 209–223.
- Hope, J., . *Biobazaar: The Open Source Revolution and Biotechnology*. Harvard University Press.
- Hsu, J., Wang, X., 2005. On welfare under cournot and bertrand competition in differentiated oligopolies. *Review of Industrial Organization* 27, 185–191.
- Johnson, J.P, 2002. Open source software: Private provision of a public good. *Journal of Economics & Management Strategy* 11, 637–662.
- Lerner, J., Pathak, P.A., Tirole, J., 2006. The dynamics of open-source contributors. *The American Economic Review* 96, 114–118.
- Lerner, J., Tirole, J., 2000. The Simple Economics of Open Source. Number 7600 in NBER Working Paper Series.
- Lerner, J., Tirole, J., 2002. Some simple economics on open source. *Journal of Industrial Economics* 50, 197–234.
- Llanes, G., de Elejalde, R., 2009. Industry Equilibrium with Open Source and Proprietary Firms. *Harvard Business School Working Papers* 09-149. Harvard Business School.
- Maurer, S.M., 2008. Open Source Biology: Finding A Niche (Or Maybe Several). Technical Report 2.
- Maurer, S.M., Scotchmer, S., 2006. Open source software: the new intellectual property paradigm, in: Hendershott, T. (Ed.), *Handbook of Economics and Information Systems*. Elsevier, pp. 285–319.
- Osterloh, M., Rota, S., von Wartburg, M., 2001. Open Source - New Rules in Software Development. Technical Report.
- Pénin, J., Wack, J.P, 2008. Research tool patents and free-libre biotechnology: A suggested unified framework. *Research Policy* 37, 1909 – 1921.



- Quah, D., 2003. Digital Goods and the New Economy. CEP Discussion Papers 563. London School of Economics. London.
- Roosendaal, A., 2007. Biomedics and creative commons: A perfect match? *International Journal of Intellectual Property Management* 1, 316 – 324.
- Rossi, C., Bonaccorsi, A., 2006. Intrinsic motivations and profit-oriented firms in open source software: Do firms practise what they preach?, in: Schröder, J.B., H., P.J. (Eds.), *The Economics Of Open Source Software Development*. Elsevier, pp. 84–109.
- Rossi, M.A., 2006. Decoding the free/open source software puzzle: A survey of theoretical and empirical contributions, in: Bitzer, J., Schröder, P. (Eds.), *The Economics Of Open Source Software Development*. Elsevier, pp. 15–55.
- Schiff, A., 2002. The economics of open source software. a survey of the early literature. *The Review of Network Economics* 1, 66 – 74.
- Schmidtke, R., 2006. Private Provision of a Complementary Public Good. Technical Report.
- Singh, N., Vives, X., 1984. Price and quantity competition in a differentiated duopoly. *RAND Journal of Economics* 15, 546–554.
- Sutton, J., 1997. One smart agent. *RAND Journal of Economics* 28, 605–628.
- Symeonidis, G., 2003. Comparing cournot and bertrand equilibria in a differentiated duopoly with product r&d. *International Journal of Industrial Organization* 21, 39–55.
- Verani, S., 2006. Open Source Development in a Differentiated Duopoly. Technical Report.
- Weber, S., 2004. *The Success Of Open Source*. Harvard University Press.

## A. Appendix

If  $\phi > \phi_{\bar{x}}^{\text{OS}}$ , then there is always only one unique equilibrium in OS development, and furthermore CS output decreases  $x^{\text{OS}*}$ . Figure A depicts this



with an example of  $z = 2$  OS-firms: if  $x^{\text{CS}}$  is moderate, OS firms develop  $x^{\text{OS}} > 0$  (left hand), while if there is much  $x^{\text{CS}}$ ,  $x^{\text{OS}}$  production is driven to zero (right hand).

For  $\phi < \phi_{\bar{x}}^{\text{OS}}$ , the situation is more complex. In this case the slope of the OS-reaction functions (with respect to the other firms' OS) are greater than one. This implies that multiple equilibria can exist. Again, we illustrate this graphically for the case of  $z = 2$ . If  $x^{\text{CS}} = 0$  or low enough, then the OS firms together develop the cutoff, see the left hand of figure 7. But if  $x^{\text{CS}}$  is high enough, then the OS-reaction functions are shifted so that multiple equilibria exist. The right hand side of figure 7 depicts such a situation. Without additional assumptions, each of the three equilibria are plausible. Hence, either the OS firms develop the cutoff, or the equilibrium  $0 < X^{\text{OS}} < \bar{x}$  establishes, or the OS firms develop no code at all.

Figure 7: OS-Equilibrium and the Impact of  $x^{cs}$  if  $\phi < \phi_{\bar{x}}^{os}$

