UNIVERSITY OF CALIFORNIA, BERKELEY
GOLDMAN SCHOOL OF PUBLIC POLICY

GSPP10-001

The New (Commercial) Open Source: Does It Really Improve Social Welfare?

Sebastian v. Engelhardt

Friedrich-Schiller-University Jena, Department of Economics and Business Administration. Email: sebastian.von.engelhardt@uni-jena.de

Stephen M. Maurer

University of California, Goldman School of Public Policy. Email: smaurer@.berkeley.edu

Abstract

We discuss welfare and various policy interventions for mixed ICT markets where firms use either 'open source' (OS) or 'closed source' (CS) business models. We find that the existence of OS business models improves social welfare compared to all-CS industries by letting firms share costs and avoid duplication. However, code sharing also establishes a de facto quality-cartel that suppresses OS firms' incentives to invest. Competition from CS firms weakens this cartel and improves welfare. That said, market forces alone provide too little CS competition. We find no support for various government interventions based on tax breaks for OS-based firms and pro-OS procurement preferences by government. However, policies that directly target the supply of OS code have a positive impact.

Keywords: open source, commercial open source, non-cooperative R&D, procurement preferences, government interventions
JEL: H25, L17, O34, O38

This Version: August 2012

1. Introduction

Despite differences in detail, the number of conceptually distinct incentives (e.g. patents, prizes, grants, contract research) that society uses to promote innovation is remarkably small (see Scotchmer, 2004). Against this background, the emergence of fundamentally new ‘open source’ (OS) methods for producing software in the 1990s surprised and delighted observers. Furthermore, OS seemed to avoid proprietary or ‘closed source’ (CS) software’s worse feature—charging consumers a royalty for information that could theoretically be distributed at zero cost. This made it natural to ask whether OS could drastically improve welfare compared to CS.

For a long time, this question was largely theoretical. Early OS collaborations were almost entirely centered on non-monetary incentives like intrinsic motivations (fun, altruism), education, signaling, and reputation (Lerner and Tirole, 2002; Ghosh et al., 2002b; Lakhani and Wolf, 2005; David and Shapiro, 2008). This made it impossible for governments to influence OS production using policy levers (e.g. taxes, subsidies) that are based on financial incentives (Schmidt and Schnitzer, 2003; Evans and Reddy, 2003; Lee, 2006). Moreover, OS programmers’ need for, say, reputation was limited,[1] and without a price signal, consumers had no direct way to influence the supply of OS software. In the first years of the 21st Century, then, one might have expected OS production to level off.

In fact, the opposite happened. Indeed, the number of worldwide OS projects grew nine-fold between 2001 and 2007 [2] and has continued to grow exponentially since then (Riehle, personal communication). But this was only possible because OS itself changed. As Lerner and Schankerman (2010)’s massive survey reveals, the old pattern of “a widely dispersed pool of voluntary contributors” has given way to projects in which “corporations increasingly invest large amounts of money to finance open source development, both in terms of direct finance to other companies and through paying their employees to engage in such activity.”[3] Several of these OS projects (e.g. Linux, Android), have become household names representing billion dollar

[1]As Fershtman and Gandal (2004) reports in the case of Apache, OS volunteers only need so much reputation. Many volunteers drastically reduce code contributions as soon as collaboration organizers promote them to a higher status.

[2]Deshpande and Riehle (2008) report exponential growth of OS projects in their representative worldwide sample of OS projects.

[3]Similarly , Barnett (2011) writes that “[s]tandard characterizations of OSS develop-

investments by dozens of corporations and hundreds of professional programmers. Lerner et al. (2006) report that firm participation is strongly correlated with the largest and fastest growing projects.[4]

Inevitably, this new world brings fresh policy challenges. First, OS is far more central to the economy than it was before. Second, OS is now largely commercial. This technology sharing between firms raises obvious, if unfamiliar, issues for competitions policy. Third, and most importantly, there are now dozens of markets in which CS and OS software compete head-to-head (Harvey, 2010). This means that it is no longer realistic to analyze pure-CS or pure-OS markets in isolation. Instead, any complete theory (let alone policy) must ask when mixed-OS/CS markets are possible, whether and to what extent they are desirable, and what government can do to promote and improve them. The good news is that politicians understand this and routinely stress the need for balanced OS/CS "ecosystems." The bad news is that there is still no clear concept of what such a balance consists of, let alone how to know when an appropriate OS/CS mix has been achieved.

This article presents a broad framework that policymakers can use to understand and manage commercial OS, both in isolation and as part of a mixed OS/CS ecosystem. Like all models, we do not represent every last detail of the activity. Instead, we try to capture the underlying economic logic that unites the great majority of OS business models. This kind of deliberately generic approach is essential if policymakers are to frame global rules (e.g. tax policy) for OS as a whole. The only real question is whether our model captures the main features of the problem. We argue below that the great majority of commercial OS activity is based on selling proprietary complements and that global interventions can be safely designed on this basis.

---

ment as the spontaneous coordination of [...] ideologically motivated volunteers [...] do not accurately describe at least the most successful applications in the current market.". Lohr (2009) reports that the days when OS was driven by sharing rather than profits "are long gone". Survey research has similarly detected a secular shift in OS motivations away from hobbyist-based production to a culture dominated by paid professionals working on company time. (Ghosh et al., 2002a,b).

[4]Indeed, the unthinkable has happened: At least one leading OS collaboration (Eclipse) receives more software deposits during office hours than it does on weekends (Weingarten, personal communication).

2. Literature

The idea that commercial firms are prepared to fund shared research for unsentimental, profit-maximizing reasons is familiar from the (non-)cooperative R&D literature. This provides a solid foundation for analyzing commercial OS. At the same time, firms have spent much of the past decade experimenting with, and often discarding, specific business models. Much of the empirical literature is devoted to cataloging this activity. This has encouraged researchers to focus on differences in detail and delayed efforts to find underlying commonalities. On the theory side, the analysis has been similarly fragmented. Until recently, most workers concentrated on modeling relatively narrow phenomena that OS collaborations could be expected to display under plausible circumstances. The challenge now is to build broader theories that build on and incorporate these insights.

This section reviews what scholars have learned about commercial OS. Section 2.1 reminds the reader of how OS fits into the broader non-cooperative R&D literature. Section 2.2 reviews recent efforts to catalog the many different business models that OS firms have experimented with since the mid-1990s. Despite superficial differences, we argue that almost all of these examples depend on bundling an OS product with one or more privately owned complements. Section 2.3 reviews economists' efforts to model commercial OS. These typically analyze investment decisions within the framework of two-stage duopoly games. Finally, Section 2.4 reviews the theory and practice of government OS policy. Here, recent proposals almost always stress the need for balanced approaches that treat OS and CS firms as a single combined "ecosystem". The problem, for now, is that politicians have said almost nothing about what this balance consists of. We argue that further understanding will require an explicit, deliberately generic model. This motivates the rest of our article.

2.1. Cooperative and Non-cooperative R&D, and Spillovers

Commercial OS development can usefully be analyzed as an example of technology sharing (Lerner and Tirole, 2005) or pooled R&D (West and Gallagher, 2006). This links the phenomenon to (non-)cooperative R&D models, most prominently represented by the work of d'Aspremont and Jacquemin (1988). However, most of this work focuses on cost-reducing R&D. By comparison, OS/CS development is almost always directed toward improving the quality of ICT products. This locates OS within the sub-literature that analyses how R&D affects vertical product differentiation (e.g. Motta, 1992; Rosenkranz, 1995; Deroian and Gannon, 2006).

In what follows, we analyze the the case where firms choose between two different non-cooperative R&D strategies within a single market. These two different non-cooperative R&D strategies imply dramatically different spillovers while. This extends the literature since previous analyses have almost always assumed that spillover rates differ, if at all, based on whether firms pursue cooperative or non-cooperative R&D (Kamien et al., 1992; Choi, 1993; Miyagiwa and Ohno, 2002; Hinloopen, 2003).

The fact that the firms in our model choose different levels of spillovers (OS vs. CS) distinguishes our research questions also from those dealing with general, industry-wide spillovers.[5] More related to our topic is Fershtman and Gandal (2011) who analyze the direct and indirect knowledge spillovers between OS projects and OS developers. Since we focus on the difference between OS and CS, we abstract from these details and treat OS spillovers as a single homogeneous activity.

2.2. Empirical Studies of OS Business Models

Commercial OS firms have experimented with—and frequently discarded—a wide variety of business models since the 1990s. Given this turmoil, much of the early literature focused on detailed case studies of individual collaborations (Välimäki, 2003). By the early 2000s, however, scholars began to locate these individual examples within broader taxonomies (Ghosh et al., 2002a; Fink, 2002; Daffara, 2007). This work provides a solid starting point for workers hoping to construct the kinds of broadly generic models that tax policy and other global interventions require.

Table 1 summarizes this literature. The core of this list (Items 1 - 9, 16) adopts Daffara's state-of-the-art taxonomy (Daffara, 2007, 2009). In order to ensure completeness, we have conservatively added six additional categories found elsewhere in the literature.

[5]There is an extensive literature on how knowledge spills over via different channels like labor mobility or face-to-face communications and the like, giving reasons to local/geographical effects and clusters (e.g. Audretsch and Feldman, 1996; Almeida and Kogut, 1999; Bresnahan et al., 2001; Dahl and Pedersen, 2004; Sorenson et al., 2006) and their role in markets for technology (Arora and Gambardella, 2010). Since the technology sharing of OS is not limited to these channels, we do not take this aspect into account.

Table 1: Commonly Cited Reasons Why Firms Contribute to OS Software

Name	Description	Motivation
1. Open Core	Firm bundles open software with proprietary plug-ins (Daffara, 2009).	Selling proprietary software.
2. Product Specialists	Firm produces open software while offering training and consulting services (Daffara, 2009).	Selling user support services.
3. Platform Providers	Firm integrates separate open source components into a single, custom-designed platform (Daffara, 2009).	Selling programming services.
4. Selection and Consulting Services	Firm evaluates and recommends proposed ICT projects. Daffara (2009) remarks that these firms produce relatively little OS code and have :"a very limited impact on (...) FLOSS communities" (Daffara, 2009).	Selling consulting and evaluation services.
5. Aggregate Support Providers	Firm provides "one-stop" support for users running OS programs adopted from multiple unrelated sources (Daffara, 2009).	Selling support services.
6. Legal Certification and Consulting	Firm manages legal issues associated with OS use. Daffara (2009) remarks that these companies seldom create OS (or even CS) software (Daffara, 2009).	Selling legal, compliance audit, and insurance services.
7. Training and Documentation	Firm offers courses, training, and manuals (Daffara, 2009).	Selling support services.
8. Indirect Revenues	"A company may decide to fund open source software projects if these create a significant revenue source for related products, not directly connected with source code or or software" (Daffara, 2009). Examples include hardware and hardware drivers (see also Rossi and Bonaccorsi, 2006).	Selling hardware and software.

Table 1 continued

Name	**Description**	**Motivation**
9. Knowledge and R&D Cost Sharing	Firm opens code so that outside developers improve and debug it. (Henkel, 2006b; Rossi and Bonaccorsi, 2006).	*see text*
10. First-Mover Advantages and Network Effects	Firm opens code to expand user network. Free code subsidizes new users, acts as a commitment strategy that limits firm's ability to raise prices in the future, and reassures users that software will be maintained (Maurer and Scotchmer, 2006).	*see text*
11. Monitoring/ Influencing OS Projects	Firm contributes OS in order to monitor and/or influence development efforts (Maurer and Scotchmer, 2006).	*see text*
12. Recruitment	Firm participates in OS products to identify, hire gifted programmers (Rossi and Bonaccorsi, 2006; Maurer and Scotchmer, 2006).	Improved Hiring.
13. Retention and Morale	Firm participates in OS to retain and promote high performance among programmers.	Employee Performance.
14. Public Relations	Firm offers OS code or participates in OS collaborations to improve its image.	Reputation.
15. Norms	Firm offers OS code or participates in OS collaborations for nonfinancial, ideological reasons (Osterloh, 2002; Rossi and Bonaccorsi, 2006).	Non-commercial Reward
16. Dual Licensing	Firm offers open version of proprietary product in order to obtain bug fixes and ideas for new product features. "Mainly used by producers of developer-oriented tools and software" (Daffara, 2009).	Bug fixes and ideas from non-commercial users.

We proceed by discussing Table 1 in detail. The core of our list, Items 1 through 8, is quickly addressed. Readers can confirm by inspection that each of these business models funds OS production through the sale of privately owned complements. In economic terms, at least, the fact that the complement consists of services (Items 2, 3, 5-7), software (Items 1, 9), or hardware (Item 9) can be analyzed as well-defined subcases.

Properly speaking, Items 9 (R&D Cost Sharing), 10 (Monitoring OS Projects) and 11 (First Mover Advantages) are not really business models at all. Instead, each is recursive in the sense that it depends on additional, unstated models to earn revenue. For example, Item 9 assumes that users will contribute to the project. To the extent that these users are commercial, we expect them to fund the activity through some other business model. Similarly, Items 10 and 11 argue that firms often participate in open source to gain strategic advantage. This, however, implies that the firm expects to extract revenue at a later date by unspecified means.

Item 12 (Recruitment) argues that firms value OS as a uniquely transparent way to identify and evaluate job candidates. To the best of our knowledge, however, no scholar has ever analyzed how much OS involvement a company needs for this purpose. Naively, even small-scale participation should be sufficient to identify capable programmers.

Items 13 through 15 (Retention, Public Relations, Norms) advance non-economic reasons for firms to invest in OS. For now, relatively little is known about the importance of these incentives in generating OS software by firms. There is some evidence, however, that firms who self-report strong values-based reasons for supporting OS invest relatively little: Rossi and Bonaccorsi (2006) report that about two-thirds (62%) of firms expressing above-average support OS provided less-than-average support. Furthermore, similar motives have been widely studied in the corporate social responsibility literature. Here, the conventional wisdom holds that firms have only limited discretion to depart from strict profit maximization and, in any case, cannot support actions that would lead to negative profits (Vogel, 2005). For this reason, we expect Items 13 - 15 to play relatively minor roles in overall OS software production.

To this point, we have argued that the great majority of OS business models can be accurately described in terms of simple complements models. Item 16 is different. Dual Licensing models typically involve cases where a single firm creates code and then offers access to users under either an OS or a CS license. The OS community generates bug reports and new ideas while the CS community provides revenue.[6] Several scholars have reported that dual licensing models have been important business models in the past. Today, however, this status seems to be eroding.[7]

We understand that the foregoing qualitative discussion is not entirely satisfactory. We would much prefer a world-wide census that ties a large, representative sample of OS software to specific business models.[8] In the meantime, however, we can refer to Daffara (2009) who reports that 93% of OS firms use business models covered by Items 1-8, where OS is combined with a complementary product. Thus, the available evidence suggests that policymakers should design tax rules and other global interventions on the assumption that OS firms support themselves by selling complements.[9]

2.3. Theories of Commercial OS

Theories of traditional (non-commercial) OS almost always ask why unpaid volunteers should invest time and effort in inventing new products. The rise of commercial OS provides a trivial answer: Firms hire programmers to do the work. But this answer begs the deeper question of why profit-maximizing firms should do such a thing in the first place. Taking their cue from the empirical literature, theorists have universally assumed that OS firms recoup their investment by selling proprietary complements. This is

[6] Comino and Manenti (2011) provide an instructive analysis. Their key insight is that the firm relies on OS users to cut costs by supplying bug reports while CS users provide revenue. This means that profits are determined by the size of each population. This can often be manipulated. For example, firms seeking to increase revenue can adopt strong licenses that discourage OS use.

[7] For example, Daffara (2009) records that dual licensing came in fourth out of ten categories considered. However, he also reports that there is no single "significant" model today and that "the most probable future" points to development consortia (Item 9) and product specialists (Item 2). As we have seen, these are both complementary products models.

[8] For now, the leading web census is due to Deshpande and Riehle (2008). Adding business model information to their data would be a massive undertaking.

[9] Alternatively, policymakers may sometimes be able to design tailored policies that distinguish OS businesses that support themselves through the sale of complements from those that do not. Businesses operating dual use license models, in particular, tend to be highly distinctive. Unlike other commercial OS collaborations, they almost always feature (a) governance structures that centralize power within a single dominant firm, (b) OS licenses that feature very strong, copyleft restrictions, and (c) little or no code production outside the dominant firm. See, e.g. Välimäki (2003).

invariably modeled using Cournot and Bertrand competition. Until recently, almost all of this work was limited to monopoly and duopoly models. This approach was mathematically convenient but still sufficiently powerful that researchers could explore topics like OS firms' use of non-commercial code or the deliberate choice of CS models as a deterrent to entry. In retrospect, however, it also obscured other issues—most notably competition by two or more OS firms against a CS rival—that simple duopoly models cannot capture.

The first commercial OS studies typically explored business models in which a commercial OS firm (e.g. Red Hat) builds its business on a preexisting body of non-commercial code.[10] For example, Bitzer (2004) presents a model in which OS firms receive software at zero cost and invest their entire R&D budget in developing complements. He finds that CS firms which write their own software can nevertheless survive in the market provided that their products are sufficiently heterogeneous. Sen (2007) similarly analyzes whether CS firms can compete against OS firms selling improved versions of existing, non-commercial OS software. He finds that both types of firms can sometimes coexist. However, when network effects create winner-take-all markets, Sen expects OS firms to drive CS rivals out of the market.

These early analyses assumed that OS firms were essentially parasitic, i.e. relied on production by non-commercial volunteers. By the mid-2000s, however, OS firms were creating their own code bases. Baake and Wichmann (2004) were the first to analyze this behavior. They constructed a Bertrand model in which two software firms decide whether to publish part of their software as OS code. They found that both firms would release significant OS software in equilibrium in order to obtain bug reports from users. However, they also noted that OS software reduced barriers to entry. Thus, incumbents would deliberately limit OS code donations short of the point where new OS firms would enter the market. Finally, Verani (2006) constructed a differentiated duopoly model to compare output in all-OS and all-CS industries. He found that firms develop more code under OS regimes when their software-based products (bundles) are close substitutes.

To the best of our knowledge, the first author to proceed beyond duopoly

[10] Relatedly, some authors asked when CS firms could earn revenue in competition with non-commercial software. (see e.g. Mustonen, 2003; Casadesus-Masanell and Ghemawat, 2006).

models was Schmidtke (2006), who constructed a Cournot model in which OS firms invest in a shared public good (e.g. OS software) to increase the value of a private complement (e.g. server hardware). Unlike previous work, Schmidtke's model can accommodate arbitrarily large numbers of OS firms. This allows him to explore the impact of entry on profits, output, and total welfare.[11] Schmidtke concludes that welfare increases with the number of firms. At the same time, his model assumes a pure-OS market. For this reason, he is unable to analyze competition between OS and CS firms.

More recently, Llanes and de Elejalde (2009) present a two stage model in which CS and OS firms compete by bundling a primary good (which can be OS or CS) with a complementary private good. In the first stage, firms decide whether they are an OS or CS firm. In the second stage, firms decide on both the quality and price of the bundle that they plan to offer to consumers. Llanes and de Elejalde find that relatively few firms choose OS when most of the bundle's value derives from the primary good. This is because OS firms find it hard to recoup investment from the open complement. However, the situation changes when the complement is roughly as valuable as the primary. Here, the cost advantage of code-sharing dominates so that all firms adopt OS. Related work by Casadesus-Masanell and Llanes (2011) extends this model to the general case where firms can choose between making all, part, or none of their modules OS. They find that firms are more willing to open modules that are substitutes to existing open-source projects. Firms open high-quality modules when CS and OS modules are incompatible, while low-quality firms are more open when modules can be combined.

The Llanes and de Elejalde (2009) model was developed independently from our work and differs in important respects. Most obviously, the authors do not investigate the impact of code-sharing on quality competition among OS firms. This is probably an artifact of their set-up, in which firms select the quality of the OS good and the price of the complement simultaneously.[12] Though mathematically convenient, we argue that this choice overlooks the fact that software R&D projects have much longer lead times than hardware

[11] Schmidtke also points out that firms may sometimes use OS to subsidize their competitors' R&D programs. This makes sense where the new technology is a strong complement that promises to increase demand for both firms' products. Henkel (2006a) similarly argues that complements can encourage reciprocal OS disclosures.

[12] We find that inserting similar assumptions into our model also suppresses quality competition effects.

pricing decisions. Moreover, the difference is important. We argue below that shared OS code suppresses quality competition and leads to underinvestment. This poses a central challenge for policymakers.[13]

Finally, we note that simple profit-maximization models are not the whole story. First, Henkel (2006a) remarks that firms' ability to share individual R&D tasks is limited. Strangely, this defect can sometimes *increase* total OS production by forcing each firm to specialize in whichever task(s) it finds most valuable. Second, the rise of OS collaborations adds a new layer of institutions that cuts across firms. Indeed, many programmers continue to participate in collaborations after changing employers. OS firms must consider this dual loyalty before investing. (Dahlander and Wallin, 2006; Henkel, 2009; Rolandsson et al., 2011). Similarly, firms' readiness to invest in OS business models can depend on whether the project grows out of an earlier, non-commercial community (Dahlander, 2007). Furthermore, firm participation has an influence on the the type of governance structure selected (West and O'Mahony, 2008) and the particular license chosen (Koski, 2005). Finally, many scholars argue that OS output is also affected by expected reciprocal behavior from the OS community (Henkel, 2006b), as well as by norms, perceived fairness, and other non-economic factors (Osterloh et al., 2001; Osterloh, 2002).

Taken as a whole, the existing theoretical literature identifies many of the policy issues that are likely to be encountered where OS and/or CS firms compete through the sale of bundled complements. This provides a solid foundation for more general models that policymakers need to evaluate global interventions like tax policy.

2.4. Government Interventions

Advocates have long argued that more OS would improve welfare. Rationales include diluting large CS firms' market power, promoting local content, boosting exports, creating new platforms for the private sector to build on, promoting e-government, and strengthening national security. Arguments

[13]The Llanes et al. models also differ from the current work in other respects. Most notably, they assume that (a) demand follows a differentiated Hotelling model, (b) the total number of potential entrants is fixed, and (c) firms can choose the intermediate option of making some, but not all of their code OS. To the best of our knowledge, none of these differences produces qualitatively different results from ours. This is reassuring since it suggests that our insights and conclusions are broader than our specific model.

for and against these assertions can be found in Comino et al. (2011), Lerner and Schankerman (2010) and Wong (2004). Similarly, Comino and Manenti (2005) argue that government intervention may be needed to overcome market failure caused by CS firms' greater ability to teach consumers about product features and how to use them. Finally, Comino et al. (2011) note that many OS advocates claim that open products are inherently cheaper, offer better quality, and lead to faster innovation. If proven, such arguments would make the case for intervention trivial.[14]

Even assuming that more OS is desirable, however, the question remains whether government can do anything to intervene. Early theories of OS—like the activity itself—focused on non-commercial incentives like altruism, reputation, and signaling. Since none of these involved financial incentives, it followed that traditional government levers based on taxation and/or subsidies could do little to influence production (e.g. Schmidt and Schnitzer, 2003; Evans and Reddy, 2003; Lee, 2006). This may explain why scholars have spent relatively little effort in studying interventions. So far, the most influential study promoting tax breaks and other pro-OS interventions is (Ghosh, 2006). However, his empirical work only focuses OS developers and OS firms without comparing them against their CS counterparts. Still other scholars have argued that government can intervene by adopting mandates or preferences that require agencies to purchase OS even in cases where CS products offer higher quality (Wong, 2004; O'Connor et al., 2005).

Scholarly doubts did not stop politicians from experimenting with procurement preferences, tax breaks, grants, and other pro-OS interventions (Lerner and Tirole, 2005; CSIS, 2008; Comino et al., 2011). Since the mid-2000s, however, governments have dramatically broadened their focus. Instead of simple OS-promotion schemes, today's politicians typically advocate "a search for business models that can profitably blend open and proprietary processes and products" (CSIS, 2008, see also; Sharpe, 2009; Buono and Sieverding, 2009). We describe these developments further in Section 6.

The rise of commercial OS collaborations removes the old objection that government can do nothing to influence OS volunteers. Today's commercial OS production is clearly amenable to taxes, subsidies, and other financial

[14] Lerner and Schankerman (2010) argue intellectual property monopoly overprices CS so that it is underused. This leads them to the opposite conclusion, i.e. that government should intervene to increase the supply of CS.

incentives. At the same time, politicians' instinct that interventions should aim for balanced OS/CS ecosystems deserves to be taken seriously. In what follows, we give content to these ideas by exploring a very general model in which OS and CS firms compete with one another.

3. A Simple Commercial OS Model

We model private sector decisions to develop OS and CS software as a two-stage game:

In Stage One, profit-maximizing firms decide how much code to develop (either as shared OS or private CS).

In Stage Two, the firms produce a complementary product whose performance depends on the code. They then sell the bundled products in markets that include two or more competitors.

Since the stage one and two products are complements, consumers shop for bundles rather than individual products. We stylize our model so that (a) while the software and complement determine the quality of the bundle, the complement's quality is exogenously given, and (b) firms engage in Cournot competition, i.e. compete on quantities (for Bertrand, see Appendix A). This allows us to identify firms' *Stage One* decisions with *quality* and their *Stage Two* decisions with *quantity*. We discuss the impact of different levels of the complement's quality in Section 7.

3.1. Stage Two: Decision on Quantity

Each firm chooses whichever level of output q_i maximizes its profits $\pi_i = p_i q_i - c_i - C$, where p is price, q is the number of bundles sold, and C is the fixed cost of developing the Stage Two product.[15] Furthermore, c_i denotes the costs of software development determined in Stage One. We model the price that consumers are willing to pay for each company's bundle by the inverse demand function[16]

$$p_i = \alpha_i - q_i - \gamma \sum_{j \neq i} q_j \tag{1}$$

[15]Without loss of generality, marginal costs in Stage Two are normalized to zero.

[16]This is derived from a utility function used by Dixit (1979) and Häckner (2000). The literature also discusses a second type of demand function which leads to similar results, see Engelhardt (2010).

The quality of firm i's bundle is given by α_i. The variable γ captures horizontal product differentiation. This can range from cases of perfect substitutes where $\gamma = 1$ ("mobile phones vs. mobile phones") to $\gamma = 0$ scenarios where products hardly compete at all ("mobile phones vs. software-driven toasters").

Our inverse demand function is particularly convenient for analyzing Cournot competition because it leads to equilibria in which prices equal quantities. Thus Stage Two competition induces each firm to supply the following quantity of goods:

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

Note that (2) has two important features. First, prices and quantities depend on the difference between the quality of firm i's bundle α_i and the quality of competing bundles. This term is weighted by θ, which indicates the degree to which firms engage in Stage One *quality* competition. We adopt $\theta = \frac{\gamma}{(2-\gamma)}$ as a convenient rescaling of our original measure of substitution γ. Second, p_i and q_i decline when $h = 2 + (n-1)\gamma$ increases. Here h—which depends on the number of competitors $(n-1)$ weighted by degree of substitution γ—captures competition without respect to quality differences. For this reason, h indexes the intensity of Stage Two *quantity* competition.

3.2. Stage One: Decision on Quality

In Stage One firms decide how much software to produce and thus, implicitly the quality of their Stage Two bundles $\alpha_i = 1 + X_i$, where X_i is the Stage One software included in the product.[17] A higher value of X_i indicates 'more' and 'better' software that makes the Stage Two bundle more valuable. On the other hand, no amount of software can increase a bundle's utility to infinity. We model this by defining an arbitrary upper limit (the 'cut-off') beyond which software production has no further impact on quality, i.e. $\alpha_i \in [1, \bar{\alpha}]$. Finally, for the sake of simplicity we stylize software development so that 'more' and 'better' software implies more code, and vice versa.

[17] A more general formulation would be $\alpha_i = \beta + X_i$, where β is the quality of the 'Stage Two'-product. For the sake of simplicity we have normalized β to one. Values different from $\beta = 1$ do not qualitatively change our results. However, OS becomes more (less) attractive for higher (lower) β, see Section 7.

In case of CS is $X_i = x_i^{\mathrm{cs}}$ and thus $\alpha_i = 1 + x_i^{\mathrm{cs}}$, In case of OS, however, is $\alpha_i = 1 + X^{\mathrm{os}} = 1 + x_i^{\mathrm{os}} + \sum x_j^{\mathrm{os}}$.

In line with the literature, we assume that Stage One software development can be approximated by a cost function with increasing marginal costs. This codifies the usual intuition that production encounters diminishing returns and is also mathematically necessary to suppress infinite code production. For simplicity we assume that this function is quadratic so that the cost of software X is given by $c = \frac{1}{2}\phi X^2$. where ϕ is the slope of an increasing marginal cost function.

In the case of CS, each firm offers $X_i = x_i^{\mathrm{cs}}$ and firm i's costs are therefore $c_i(x_i^{\mathrm{cs}}) = \frac{1}{2}\phi x_i^{\mathrm{cs}2}$. Similarly, the total cost of OS software is $c(X^{\mathrm{os}}) = \frac{1}{2}\phi X^{\mathrm{os}2}$, with $X^{\mathrm{os}} = \sum x_i^{\mathrm{os}}$. But since OS permits cost-sharing each firm only bears the fraction of costs attributable to its own code development x_i^{os}. Also, individual firm costs must sum to total OS code costs, i.e. $c(X^{\mathrm{os}}) = \sum c_i$. This yields $c_i = \frac{x_i^{\mathrm{os}}}{X^{\mathrm{os}}} c(X^{\mathrm{os}}) = \frac{x_i^{\mathrm{os}}}{X^{\mathrm{os}}} \cdot \frac{1}{2} \cdot \phi X^{\mathrm{os}2}$ for every OS firm, reflecting cost-sharing across participating firms (see also Engelhardt, 2010).

The CS and OS cost functions both reflect the conventional computer science wisdom ("Brooks' Law") that software costs scale quadratically with the number of programmers involved (Brooks, 1982). They are also consistent with empirical estimates of software costs, although some authors prefer a linear function (Dolado, 2001). Finally, we assume that firms find OS and CS code equally costly to write, i.e. that OS production has no inherent cost advantage over CS except to the extent that it allows members to share. This assumption seems justified given scholars' rudimentary knowledge of the subject (Koch, 2004; Asundi, 2005). However, Appendix B shows how the model can be extended to the case where OS and CS development have different cost functions. In this case, a systematic cost-advantage makes OS more attractive. Despite this, the main results of our model hold except that OS costs of the form $c_i = \frac{1}{2}\phi x_i^{\mathrm{os}2}$ lead to Pure-OS industries. The fact that these do not occur provides at least weak evidence that real OS and CS cost functions are comparable.

Next, we analyze Stage One decisions for Pure-CS, Pure-OS, and mixed OS/CS Industries where OS and CS firms compete.[18] In all three cases,

[18]We assume restricted ('viral') licenses that prevent firms from producing mixtures of OS and CS code in Stage One. Except for very special cases, permitting mixed production leads to Prisoners' Dilemma equilibria in which each firm consumes OS code but makes

software increases the bundled good's quality and hence increases consumer demand. This gives firms an incentive to invest in software. Furthermore, the strength of the incentive depends on marginal sales (increased revenue per added code line) which in turn depends on each firm's ability to capture the social value of its improvements in Stage Two sales (appropriability). This logic holds regardless of whether Cournot or Bertrand competition is assumed.

3.2.1. A Pure-CS Industry

Consider first an industry in which all firms practice CS. How much software is produced? In general, the answer depends on firms' strategic interactions, i.e. on how Firm A reacts to Firm B's decision to produce code. Solving for the amount of software x^{cs} that maximizes profit $\pi_i = p_i q_i - c_i - C$ leads to the following reaction function:

$$R_i^{\mathrm{cs}} = \frac{(1+(n-1)\,\theta)\left(1-\theta\sum_{j\neq i} x_j^{\mathrm{cs}}\right)}{\frac{1}{2}\phi h^2 - (1+(n-1)\,\theta)^2}, \tag{3}$$

where $\theta = \frac{\gamma}{2-\gamma}$ and $h = 2+\gamma(n-1)$. The SOC is $\phi > 2\,(1+(n-1)\,\theta)^2\,h^{-2}$.

An industry composed of n identical firms obeying (3) has the following Nash equilibrium:

$$x^{\mathrm{cs}*} = \frac{(1+(n-1)\,\theta)}{\frac{1}{2}h^2\phi - 1 - (n-1)\,\theta} \tag{4}$$

Here, code development is suppressed by intense quantity competition—i.e. the presence of the h^2 term—which reduces appropriability. Conversely, quality competition (θ) increases equilibrium output by making the numerator larger and denominator smaller.

Both h and θ depend on the substitutability of the bundles (γ), while h also depends on the number of competitors ($n-1$). The net effect of increasing n is straightforward: decreased output. However, changes in γ have a positive impact on both quantity and quality competition. Because quality and quantity competition exert opposing effects on output the net

---

no effort to supply it. (Engelhardt, 2010). Most OS collaborations use license terms to prevent mixed strategies.

effect is more complicated. Figure 1 plots closed source (x^{cs}) production as a function of γ.

For low-to-moderate values of γ CS code production is mainly determined by quantity competition (h), i.e. firms' ability to extract extra profits when quality increases (appropriability). This is highest when products have no substitutes ($\gamma = 0$ yields $h = h^{\min} = 2$) so that each firm can set monopoly prices unconstrained by competition. Appropriability steadily erodes as substitutability (γ)—and hence h—increases.

There is also a second effect determined by θ. For larger γ (and hence larger θ) bundles become closer substitutes, until even small quality differences lead to strong changes in demand for or against a particular bundle. This makes quality competition extremely important. Specifically, CS firms find themselves in a kind of Arms Race where each firm invests in quality to prevent every other firm from taking its business. This effect finally dominates appropriability for $\gamma > 1/2 \cdot \left(n - 5 + \sqrt{(n-2)n+9}\right)(n-2)^{-1}$, so that software production in a Pure-CS industry starts to rise again.

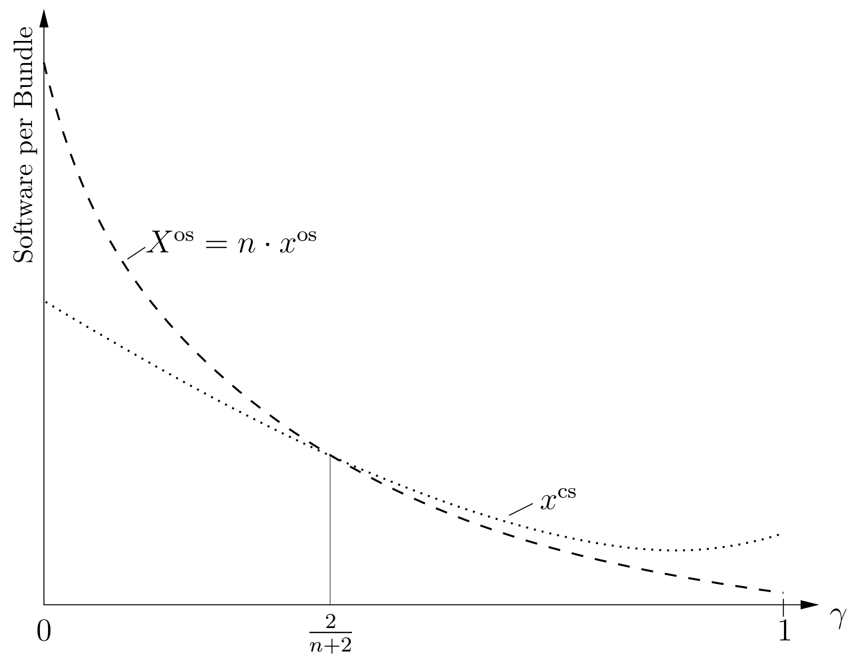
Proposition 1. *Software output in Pure-CS industries is suppressed by quantity competition but boosted by quality competition:* $\frac{\partial x^{cs*}}{\partial h} < 0$, $\frac{\partial x^{cs*}}{\partial \theta} > 0$. *As result, output falls with increasing* γ *so that the quality competition effect dominates for large* γ*:* $\frac{\partial x^{cs*}}{\partial \gamma} < 0 \; \forall \, \gamma < \frac{n-5+\sqrt{(n-2)n+9}}{2(n-2)}$*, otherwise* $\frac{\partial x^{cs*}}{\partial \gamma} > 0$.

Proof. $\frac{\partial x^{\mathrm{cs}*}}{\partial h} < 0$, $\frac{\partial x^{\mathrm{cs}*}}{\partial \theta} > 0 \; \forall \; h > 0, \; \theta \in [0,1]$ given the SOC is fulfilled. Next, $\frac{\partial x^{\mathrm{cs}*}}{\partial \gamma} \overset{!}{=} 0$ yields $\gamma = \frac{n-5+\sqrt{(n-2)n+9}}{2(n-2)}$ with $\frac{\partial^2 x^{\mathrm{cs}*}}{\partial \gamma^2} > 0$. □

3.2.2. A Pure-OS Industry

We now consider a Pure-OS industry. OS introduces two new effects. First, firms share cost. This means that for any given software output, per-firm development costs are lower for OS compared to CS. Second, OS firms share all software ($\alpha_i = 1 + \sum x^{\mathrm{os}}$). This means that a firm's decision to invest in OS software not only makes its own bundles more attractive but also—unlike the CS case—strengthens its competitors. Furthermore, the existence of shared software guarantees that no OS firm can offer better quality than any other OS firm. This suppression of quality competition implies that firms in Pure-OS industries always earn higher profits than firms in Pure-CS industries for a given number of incumbents.

Figure 1: Software per Bundle in Case of Pure-OS and Pure-CS



Solving for the amount of OS software x^{cs} that maximizes a firm's profit function yields the following reaction function:

$$R_i^{\mathrm{os}} = \frac{1 - \left(\frac{1}{4}\phi h^2 - 1\right) \sum_{j \neq i} x_j^{\mathrm{os}}}{\frac{1}{2}\phi h^2 - 1}. \tag{5}$$

where $h = 2 + \gamma(n-1)$. The SOC is $\phi > 2h^{-2}$.

Crucially—and unlike the Pure-CS case (Eqn. (3))—an OS firm's decision to produce software no longer depends on quality competition (θ) from other firms. It does, however, depend on cost-sharing.

Output decisions by n identical OS firms lead to a Nash equilibrium in which each firm produces the following amount of software:

$$x^{\mathrm{os}*} = \frac{1}{\frac{1}{4}\phi h^2 (1+n) - n} \tag{6}$$

This implies that total industry-wide output ($X^{\mathrm{os}} = nx^{\mathrm{os}}$) is

$$X^{\mathrm{os}*} = \frac{n}{\frac{1}{4}\phi h^2 (1+n) - n} \tag{7}$$

except in those cases where OS development would exceed the cut-off by delivering more code than society can use.[19]

Proposition 2. *Software output in Pure-OS industries is suppressed by quantity competition:* $\frac{\partial x^{os*}}{\partial h} < 0$*. As result, output falls with* γ*:* $\frac{\partial x^{os*}}{\partial \gamma} < 0$*.*

Proof. $\frac{\partial x^{\mathrm{os}*}}{\partial h} < 0$, $\frac{\partial x^{\mathrm{os}*}}{\partial \gamma} < 0 \ \forall \ h > 0$, $\gamma \in [0, 1]$ given the SOC and the no-cut-off condition ($h^2 \cdot \phi \cdot \frac{(n+1)}{n} > 4$, see footnote 19) are fulfilled. □

We call the intuition behind Proposition 2 the *quality-cartel effect*. We have seen that quality competition among CS firms at high γ leads to Arms Races in which firms continue to write software until rising marginal costs completely erase the profits from increased demand. OS firms, on the other hand, do not compete on quality and face no Arms Race. This suppresses code output to levels slightly lower than those that an explicit industry-wide quality cartel would set to maximize joint profits.[20]

We now compare software output in Pure-OS with Pure-CS industries.

Proposition 3. *Modest quantity competition implies high appropriability and thus* $X^{os*} > x^{cs*}$ *for Pure-OS and -CS industries having the same* n*. Since OS firms do not compete on quality (the 'quality-cartel effect'), Pure-OS industries offer less software per bundle than Pure-CS industries for* $\theta > \frac{(4-h)}{(2+h)}$*, i.e.* $\gamma > \frac{2}{(n+2)}$*.*

Proof. $X^{\mathrm{os}*} \stackrel{!}{=} x^{\mathrm{cs}*}$ yields $\theta = \frac{(4-h)}{(2+h)}$, with $X^{\mathrm{os}*} > x^{\mathrm{cs}*}$ for $\theta > \frac{(4-h)}{(2+h)}$. □

Proposition 3 expresses the fundamental difference between the Pure-OS and Pure-CS models:

- *Quantity Competition and Cost-Sharing.* As in the Pure-CS case, the amount of software produced in Pure-OS industries depends negatively on quantity competition ($h = 2 + (n-1)\gamma$) and is greatest at low γ where appropriability is high, see Figure 1. Now, however, there is a second effect. Because of shared development costs, Pure-OS industries

[19] This formally occurs where $h^2 \cdot \phi \cdot \frac{(n+1)}{n} < 4$. In practice, this condition only occurs under relatively special parameters (Engelhardt, 2010) and we ignore it in what follows.

[20] The difference stems from the fact that OS development is non-cooperative whereas a formal quality-cartel would be cooperative, see Appendix C.

are able to offer more software per bundle than Pure-CS industries as long as $\gamma < \frac{2}{(n+2)}$.

- *Quantity vs. Quality Competition.* We have seen that quality competition gradually replaces quantity competition as the most important factor in determining CS output at high θ. Pure-OS industries, however, are able to suppress quality competition through code-sharing (quality-cartel effect). This explains why Pure-CS industries deliver more software than Pure-OS industries for $\theta > \frac{(4-h)}{(2+h)}$ or, equivalently, $\gamma > \frac{2}{(n+2)}$, see Figure 1.

The balance between quantity competition and cost-sharing is consistent with Llanes and de Elejalde (2009) and Henkel (2006a)'s findings that OS business models are most profitable where quantity competition is low so that cost-sharing dominates. However, these earlier analyses do not consider the quality cartel effect in Pure-OS industries. This systematically reduces the amount of OS output that would otherwise be expected from a simple balance of appropriability and cost-sharing.

3.2.3. A Mixed OS/CS Industry

Finally, consider the case where both OS and CS firms exist. Solving for the strategic interaction within each group of firms (the SOCs are $\phi > 2\left(1+(n-1)\theta\right)^2 h^{-2}$ and $\phi > 2\left(1+r\theta\right)^2 h^{-2}$) yields equation (8), and (9) respectively

$$x^{\text{cs}} = \frac{\left(1+(n-1)\theta\right)\left(1-z^2\theta x^{\text{os}}\right)}{\frac{1}{2}h^2\phi - \left(1+(n-1)\theta\right)\left(1+z\theta\right)} \tag{8}$$

$$x^{\text{os}} = \frac{\left(1+r\theta\right)\left(1-\theta r x^{\text{cs}}\right)}{\frac{1}{4}\phi h^2\left(1+z\right) - z\left(1+r\theta\right)^2} \tag{9}$$

where z is the number of OS firms and r is the number of CS firms so that $n = r + z$. Now, however, CS and OS firms react to each other. The overall Nash-equilibrium is thus a simultaneous solution of these two functions. As before, we exclude cases where OS would exceed the cut-off by delivering more code than society can use.[21]

[21]Formally, we restrict our analysis to $\phi > 4 \cdot \frac{z}{(z+1)} \cdot \frac{(1+r\theta)^2}{h^2}$, see Engelhardt (2010).

Proposition 4. *In a mixed industry with n firms, $X^{os*} = z \cdot x^{os*}$ has an inverse U-shape over $\omega = \frac{z}{n}$, since each OS firm c.p. provides more (less) code the lower (higher) the proportion of OS firms in the market: $\frac{dx^{os*}}{d\omega} < 0$.*

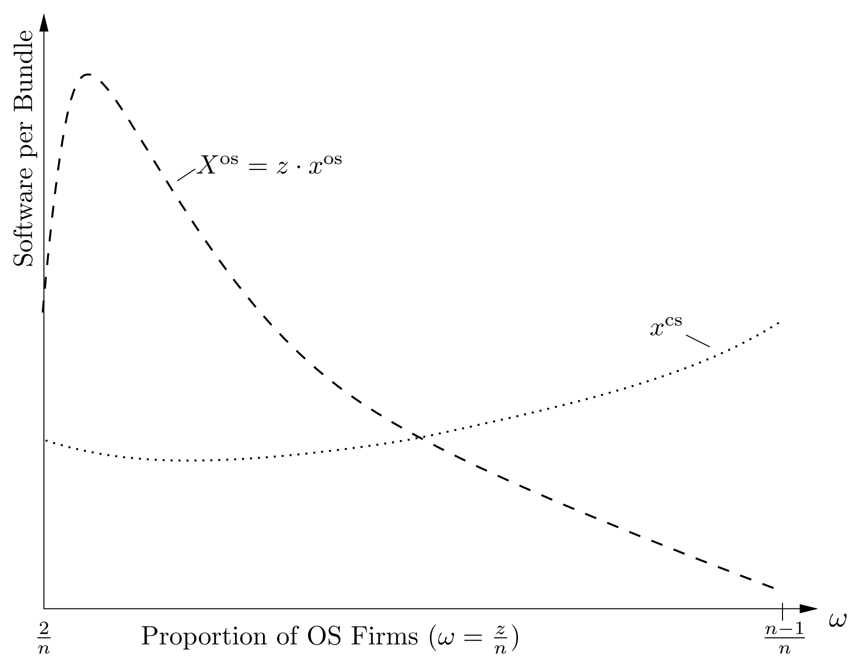
Proof. $X^{\mathrm{os}*} = z \cdot x^{\mathrm{os}*}$. $\frac{dx^{\mathrm{os}}}{d\omega} < 0$ given the SOC and the no-cut-off condition (see footnote 21) are fulfilled. Obviously, $\frac{\partial z}{\partial \omega} = n > 0$ □

Proposition 4 has the following background:

As before, each CS firm competes on quality against its competitors. The situation is different for OS firms. While OS firms still do not compete on quality among themselves, the presence of CS firms prevents OS firms from cartelizing around a low level of quality. Instead, OS firms must compete on quality against CS firms. This makes the ability to share costs more valuable to OS firms and increases OS output.

Because of these interactions, the detailed behavior of a mixed OS/CS industry depends on the ratio of OS to CS firms $\omega = \frac{z}{n}$, see Figure 2. Where OS firms are a small minority, they face strong quality competition from CS firms. This encourages them to use their cost-sharing advantage to produce large amounts of OS software. For low ω, increased opportunities for cost

Figure 2: Software per Bundle in a Mixed Market



sharing (more OS firms) continue to dominate diminished quality competition until X^{os} peaks. Thereafter, the increasing OS cartel effect ($\frac{dx^{\text{os}*}}{d\omega} < 0$) dominates so that total OS code production falls. For large ω the OS quality-cartel is so strong that OS firms produce very little code.

Because CS firms do not share costs, they cannot match the maximum potential software output that OS firms can achieve. As long as OS production is high, therefore, CS firms will specialize in selling low-quality bundles at a low price. The situation is reversed as OS productions declines. As a result, CS firms replace OS firms as the industry's high-quality, high-priced providers above a certain threshold of ω.

4. Welfare Implications

We now know how much software a Pure-OS, Pure-CS, or Mixed-OS/CS industry produces. This allows us to calculate firm profits and consumer utility, which in turn enables us to analyze welfare. More specifically, producer surplus is given by total industry profits, and consumer surplus for differentiated oligopolies is given by $\frac{1}{2}\left[(1-\gamma)\sum q_i + \gamma(\sum q_i)^2\right]$ (see Hsu and Wang, 2005). This yields the following general welfare function:

$$W = \frac{1}{2}\left[(1-\gamma)\sum q_i + \gamma\left(\sum q_i\right)^2\right] + \sum \pi_i \tag{10}$$

4.1. Pure-OS vs. Pure-CS

We begin by comparing welfare under a Pure-OS regime against the Pure-CS case. The prices and quantities of Pure-OS and Pure-CS are given by $q = p = \frac{(1+nx^{\text{os}*})}{h}$ and $q = p = \frac{(1+x^{\text{cs}*})}{h}$ respectively. The *difference* in welfare between a Pure-OS and a Pure-CS world ($\Delta W_n = W_n^{\text{os}} - W_n^{\text{cs}}$) is given by

$$\Delta W_n = \frac{n}{2}\left[\underbrace{(1+h)\,\frac{(1+nx^{\text{os}*})^2 - (1+x^{\text{cs}*})^2}{h^2}}_{\textit{quality difference}} - \underbrace{\phi\left(nx^{\text{os}*2} - x^{\text{cs}*2}\right)}_{\textit{cost difference}}\right]. \tag{11}$$

The welfare difference consists of two components. The first is the *quality difference* and the second is the *cost difference* between Pure-OS and Pure-CS. (For a further description of our welfare functions see Appendix D) For $\gamma = \frac{2}{(2+n)}$, Pure-OS and Pure-CS systems produce the same amounts of code ($X^{\text{os}} = x^{\text{cs}}$, Proposition 3, see Figure 1) so that the quality difference is zero.

Here, the remaining cost difference term makes OS welfare superior. This is because OS firms can share code whereas each CS firm must create its own code base *de novo*. This wasteful duplication of effort is variously described as "business stealing" and "me-too products" in the literature (Henkel and Hippel, 2005).

Obviously, Pure-OS retains its cost advantage for any $\gamma > \frac{2}{(2+n)}$ (i.e. for all $X^{\text{os}} < x^{\text{cs}}$). But for $\gamma < \frac{2}{(2+n)}$ Pure-OS industries produce more code than Pure-CS (Proposition 3, see Figure 1) and therefore incur higher costs. Despite this, cost sharing still makes OS cost superior in many cases:

Proposition 5. *For $\phi > \frac{n}{n+1}\frac{\sqrt{n}+1}{\sqrt{n}}$ any Pure-OS industry has a cost advantage over Pure-CS.*

Proof. If Pure-OS has a cost advantage at $\gamma = 0$ then this is also true for all other γ since $X^{\text{os}} - x^{\text{cs}}$ is maximized at $\gamma = 0$. Substituting $\gamma = 0$ in (4) and (7), applying the results to (11) and setting this equal to zero yields the boundary $\phi = \frac{n}{n+1}\frac{\sqrt{n}+1}{\sqrt{n}}$. □

Proposition 5 leads to the following statement:

Proposition 6. *For $\phi > \frac{n}{n+1}\frac{\sqrt{n}+1}{\sqrt{n}}$ and $\gamma < \frac{2}{(2+n)}$, any Pure-OS industry is welfare superior to Pure-CS.*

Proof. $\gamma < \frac{2}{(2+n)}$ implies $X^{\text{os}} > x^{\text{cs}}$ and therefore that Pure-OS has a quality advantage; $\phi > \frac{n}{n+1}\frac{\sqrt{n}+1}{\sqrt{n}}$ implies a cost advantage. □

Furthermore, the SOCs and no-cut-off boundary guarantee that $\phi < \frac{n}{n+1}\frac{\sqrt{n}+1}{\sqrt{n}}$ is only relevant where n is small. Numerical calculations of these cases show that the cost effect is never large enough to overcome the Pure-OS case's quality and cost-sharing advantages. For this reason, Pure-OS industries remain preferable to Pure-CS industries in our model for all $\gamma < \frac{2}{(2+n)}$. Significantly, this statement does not depend on ϕ and therefore holds regardless of detailed assumptions about how quickly the marginal cost of software production increases.

For $\gamma > \frac{2}{(2+n)}$ Pure-OS industries always cause fewer costs. But since $X^{\text{os}} < x^{\text{cs}}$ Pure-CS delivers higher quality. At first, OS's welfare-superiority erodes with increasing γ, until for moderate large γ CS's quality advantage dominates the cost advantage of shared OS production so that CS also delivers superior welfare. Finally, we have seen that code output in Pure-CS industries increases sharply for very intense quality competition (high θ). This can produce such large cost increases that a Pure-OS industry is once again welfare-superior.

Figure 3 summarizes these results.[22] Note in particular that OS is welfare-dominant in highly concentrated industries (low values of n), limited substitutability (low values of γ), and situations where both n and γ are moderate, hence where $h = 2 + (n-1)\gamma$ is small. Furthermore, OS is also welfare-dominant for γ close to one.

Proposition 7. *Pure-OS industries are welfare superior to Pure-CS industries for (a) low quantity competition (high appropriability), and (b) very intense quality competition.*

Proof. The proposition was confirmed by numerical calculations for different values of ϕ, taking into account the SOCs and no-cut-off condition, see also Appendix E. □

4.2. Mixed OS/CS-Industries

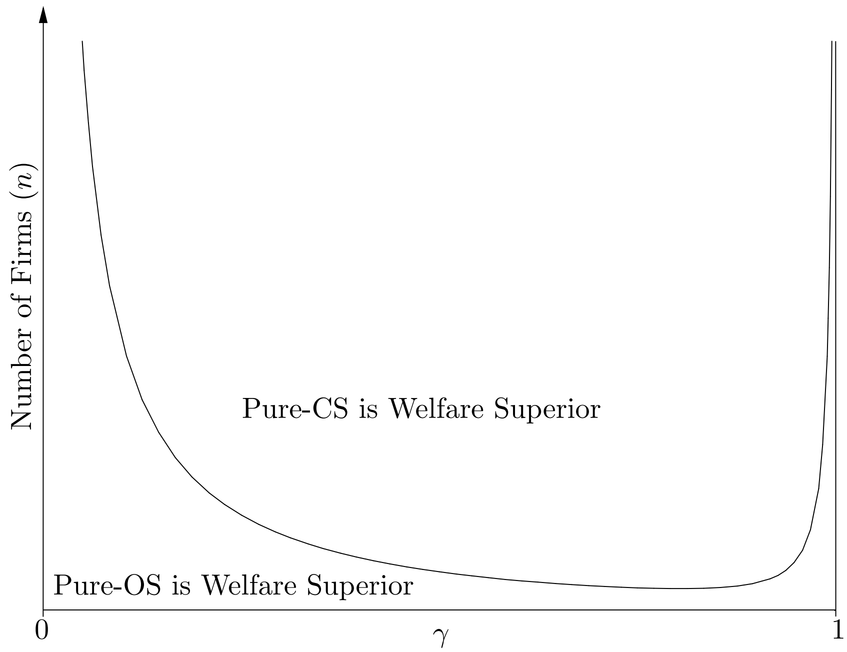
We now extend our welfare analysis to include arbitrary mixes of OS and CS firms where the proportion of OS firms is given by $\omega = \mathrm{z/n}$. We use (10) to calculate welfare for each pair (ω, γ), and compare this against our results for Pure-OS and Pure-CS industries. (For details on the welfare function see Appendix D). Figure 4 depicts typical results.[23] Figure 4 depicts when a Pure-CS, a Pure-OS or a mixed state is welfare-superior. While the figure contains the typical cases, in small n markets, some regions can disappear, see Appendix F.

In some cases, pure states offer higher welfare than mixed ones. This is reflected in the two smallest regions of the graph which are, in effect, much-shrunken versions of the Pure-OS and Pure-CS states depicted in Figure 3. First, consider the high ω region where OS firms greatly outnumber CS firms. Here, Pure-CS states are welfare-superior to mixed states for the same reasons that they dominate Pure-OS states. Second, Pure-OS is welfare-superior to mixed states in low ω/low γ cases where CS firms greatly outnumber OS

[22] The figure is based on $\phi = 2$, but results do not change significantly for values different than $\phi = 2$, see Appendix E.

[23] This is based on the example of $n = 100$ firms and $\phi = 2$. We have confirmed by careful variations that this is a useful, typical example, see also Appendix G.

Figure 3: Welfare Superiority of Pure-OS vs. Pure-CS



firms and products have low substitutability. Here, OS firms can recover their investments even without intellectual property protection. As a result, cost-sharing dominates the welfare analysis so that Pure-OS states become superior.

The largest region consists of mixed states that are welfare-superior to the corresponding Pure-OS and Pure-CS states. Moreover, such welfare superior mixed states exist for all values of $\gamma > 0$.[24]

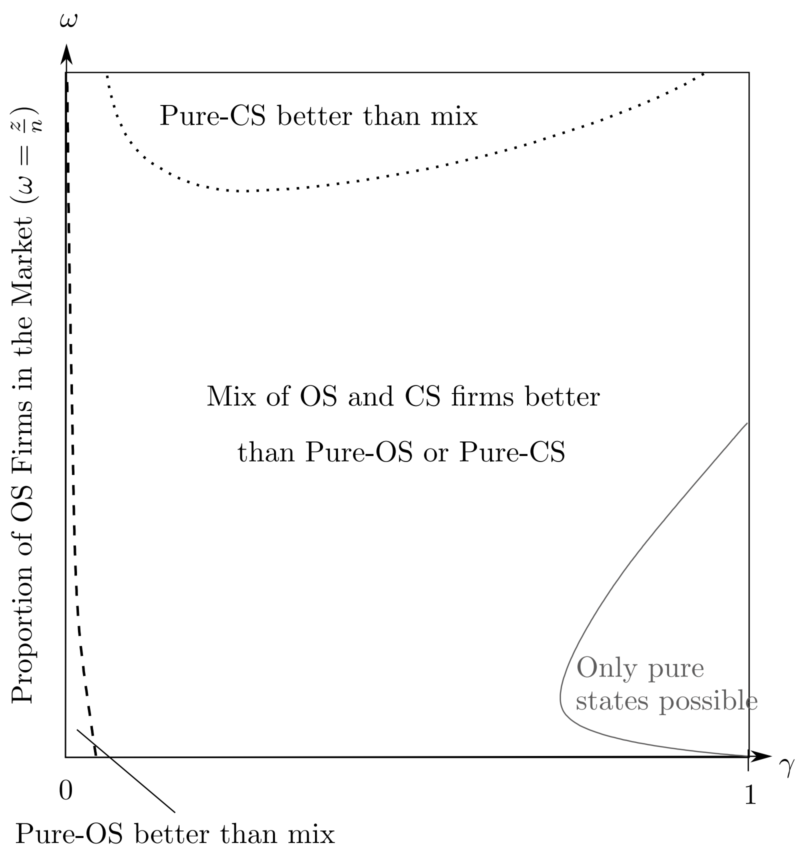
Proposition 8. *For any given $\gamma > 0$ there exists a mixed state with $0 < \omega < 1$ that is welfare superior to the corresponding Pure-CS and Pure-OS state.*

Proof. The proposition was confirmed with multiple numerical calculations for values of ϕ ranging between 1 and 20, and n ranging between 4 and $1,000,000$, taking into account the SOCs and no-cut-off condition. See also Appendix F and Appendix G. $\square$

Welfare differs within the mixed regions. While producer surplus increase

[24]There is also a region where only pure states are possible. For details on the conditions and the property of corner solutions see Engelhardt (2010).

Figure 4: Welfare-Comparison of Pure vs. Mixed Cases



in ω and reaches its maximum at $\omega = 1$, consumer surplus has an inverse U-shape. The shape of consumer surplus is driven by the quality of the bundles available in the market and mainly shaped by the inverse U-shape of X^{os} over $\omega = \frac{z}{n}$ in mixed markets (see Proposition 4 and Figure 2). Consumer surplus is maximized by mixing a few high-quality, high-priced OS-bundles with many low-quality, low-priced CS-bundles. Thus, total welfare is maximized when OS firms select high output levels in response to quality competition from a relatively large number of CS firms. We will come back to this in the following section.

5. Free Entry Market Equilibrium and Welfare

We now know when Pure-OS, Pure-CS or Mixed OS/CS provide the most welfare. Here, we explore the extent to which markets with free entry actually deliver these outcomes. Except for Baake and Wichmann (2004) and Schmidtke (2006), previous contributions start from the assumption that industry size is fixed (e.g. Casadesus-Masanell and Llanes, 2011; Llanes and de Elejalde, 2009; Henkel, 2006a). Absent compelling empirical evidence, however, ignoring entry seems artificial. Furthermore, the proportion of OS firms ($\omega = \frac{z}{n}$) and welfare implications of these models depend on initial assumptions about the number of firms. Leaving industry size a free parameter limits their predictive power.

We allow for free entry and insert a new Stage Zero into our model. Firms in Stage Zero decide whether or not to enter and, if so, whether to enter as CS or OS firms. This decision is followed by Stages One and Two as before and the extended game is solved by backward induction. Since we have already solved Stages One and Two, this section focuses on Stage Zero.

5.1. Stable Proportion of OS Firms in a Mixed OS/CS Industry

In order to be stable against entry or exit, a mix of CS and OS firms must satisfy the following conditions: (a) incumbent OS firms earn profit ≥ 0, (b) incumbent CS firms earn profit ≥ 0, and (c) would-be additional OS and CS firms *cannot* earn profit ≥ 0 by entering the market. For large n this converges to the condition[25]

$$\pi^{\text{os}} = p^{\text{os}} \cdot q^{\text{os}} - c_i^{\text{os}} - C = \pi^{\text{cs}} = p^{\text{cs}} \cdot q^{\text{cs}} - c_i^{\text{cs}} - C = 0.$$

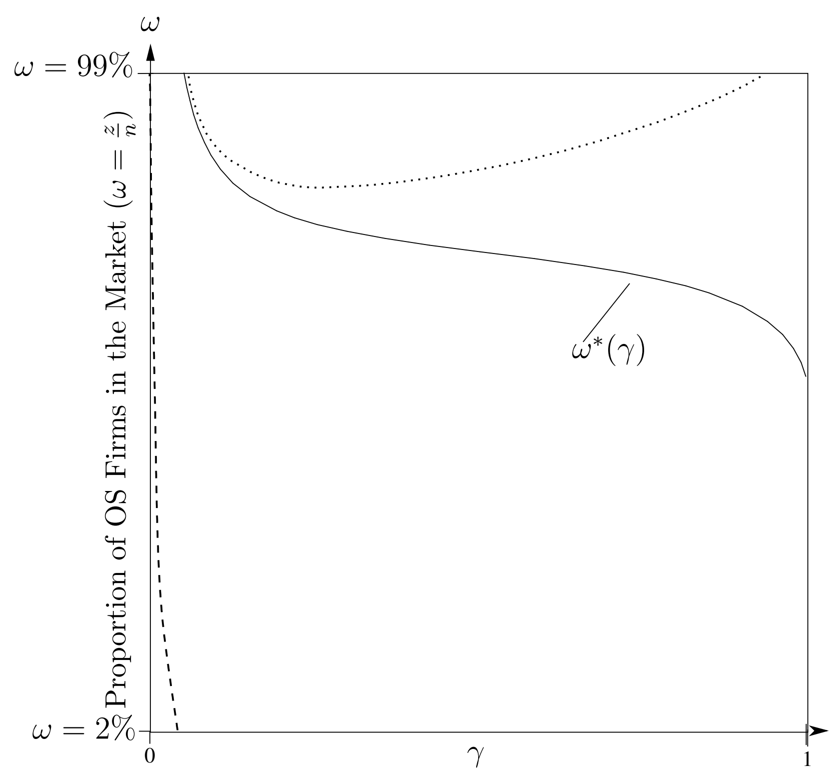
Significantly, the derived necessary condition $p^{\text{os}} \cdot q^{\text{os}} - c^{\text{os}} = p^{\text{cs}} \cdot q^{\text{cs}} - c^{\text{cs}}$ depends solely on γ, ω, n and ϕ. This allows us to calculate the stable proportion of OS firms $\omega^*(\gamma, \phi, n)$. Figure 5 depicts the stable proportion of OS firms as a function of γ, parameterized by certain ϕ and n: $\omega^*(\gamma)$. (Appendix G plots for different combinations of n and ϕ.) Figure 5 also shows the two areas where Pure states are welfare superior. The reader can confirm by inspection that $\omega^*(\gamma)$ always provides more welfare than the corresponding Pure cases. This result holds in general:

[25]The situation for small n is conceptually similar. Since the number of CS and OS firms is an integer, the calculated zero-profit ω can fall between two realizable values so that two no-entry values exist. This makes the analysis more complicated but does not change our general conclusions.

Proposition 9. *Assumed free entry and exit, any $\omega^*(\gamma, \phi, n)$ is welfare superior to the corresponding Pure-CS and Pure-OS state.*

Proof. The proposition was confirmed by multiple numerical calculations for values of ϕ ranging between 1 and 20, and n ranging between 10 and $1,000,000$, taking into account the SOCs and no-cut-off condition. $\square$

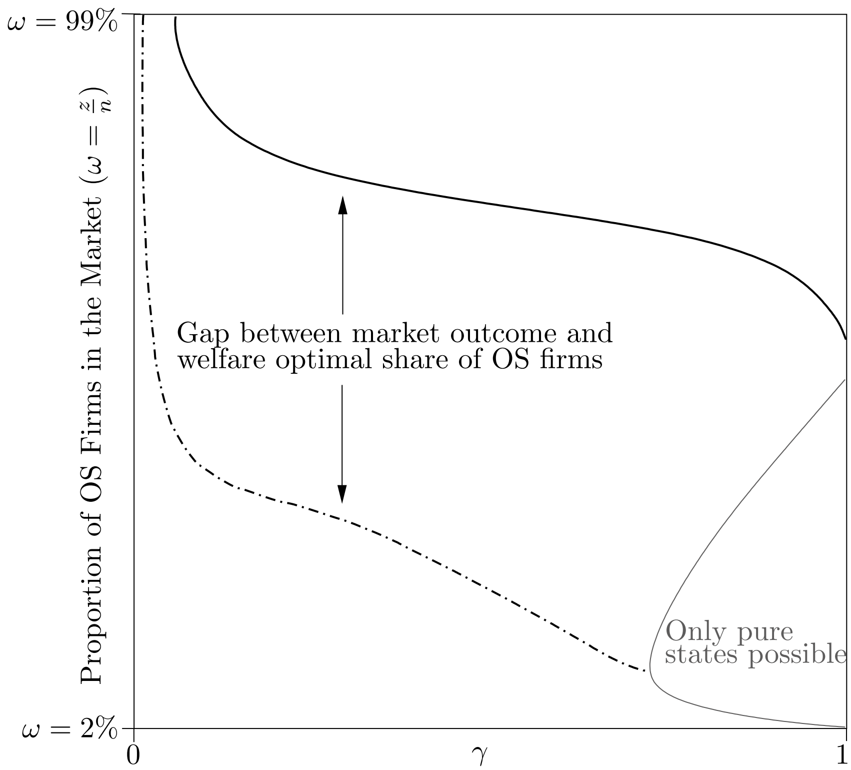
Figure 5: Entry-Resistant Proportion of OS Firms



Furthermore, $\frac{\partial \omega^*}{\partial \gamma} < 0$, i.e. the proportion of OS firms declines in industries where Stage Two products are close substitutes so that competition is high (see Figure 5). In any ω^* OS firms offer lower quality than CS firms, while each CS firm sells more bundles—i.e., has a larger individual market share—than any OS firm, see also Section 7.

We now calculate the welfare-optimal mix of OS and CS firms. Solving $\frac{\partial W}{\partial \gamma} = 0$ for ω (taking into account that x^{os}, $x^{\text{cs}} \geq 0$ and π^{os}, $\pi^{\text{cs}} = 0$) yields the welfare optimal proportion of OS firms denoted by $\hat{\omega}$. Figure 6 shows that the entry-stable proportion $\omega^{\star}$ (solid line) requires far more OS firms than the number needed to optimize welfare (dash-dotted line). This result is robust for a wide range of ϕ and n, see Appendix G.

Figure 6: Mixed Industry: Welfare Optimality vs. Stable Proportion



Proposition 10. *Compared to the welfare optimal proportion $\hat{\omega}$, there are too many OS firms in the mixed market equilibrium: $\hat{\omega} > \omega^{*}$.*

Proof. The proposition was confirmed by multiple numerical calculations for values of ϕ ranging between 1 and 20, and n ranging between 10 and $1,000,000$, taking into account the SOCs and no-cut-off condition. See also Appendix G. □

The intuition behind this is the following. We saw in the previous section that welfare is maximized at low ω where OS firms select a high quality in order to meet strong quality competition by a large number of CS firms (p.

27). However, this situation is not stable against further entry since code-sharing and the quality-cartel advantage makes incumbent OS firms more profitable than incumbent CS firms. This induces further OS entry until the situation stabilizes at ω^* so that OS firms offer lower quality than their CS rivals. The mismatch between the OS firm proportion in equilibrium (ω) and the desired welfare-optimizing $\hat{\omega}$ is a central result of this article and poses an important challenge to policymakers.

5.2. Lock In Pure States

We have seen that any entry-stable mixed market is welfare superior to the corresponding Pure-CS and Pure-OS state (Proposition 8). Suppose, though, that a particular industry starts off in a Pure OS- or CS-state. In this case, welfare improvements are bounded unless industry can transition to the mixed state. We therefore explore whether industries can become locked in against such transitions. For this purpose we allow for sequential choice in Stage Zero. The incumbents (first-mover) enter first and choose between OS and CS in Stage Zero. Next, Stage Zero's second-mover (entrants) decide whether or not to enter and, if so, whether to enter as OS or CS firms.

Pure OS Lock In. Assume that incumbents' profits in a Pure-OS industry are zero so that further OS entry is impossible. Then one can calculate whether a CS entrant would earn a non-negative profit. Figure 7 shows that industries can be locked into a Pure-OS state where substitutability and/or the number of OS incumbents is small.

Pure CS Lock In. Consider now a Pure-CS industry in which incumbents' profits are zero. As before, we assume that the Pure-CS case is unstable if OS firms can earn a non-negative profit by entering. Since OS only confers economic benefits when firms can share costs, however, at least two OS firms must enter the market to gain any advantage. We therefore consider the case where two OS companies are willing to enter the market, and would earn non-negative profits if both did so (for details see Appendix H). Figure 8 shows that while the Pure-CS state is unstable for most parameters, it is stable for (a) the concentrated, low n industries, most characteristic of Silicon Valley, and (b) industries whose products that are close substitutes.

Strategic CS Adoption by Incumbents. We now turn to the incumbents' decision in Stage Zero. The preceding analyses have shown that industries can be

Figure 7: OS Lock-In

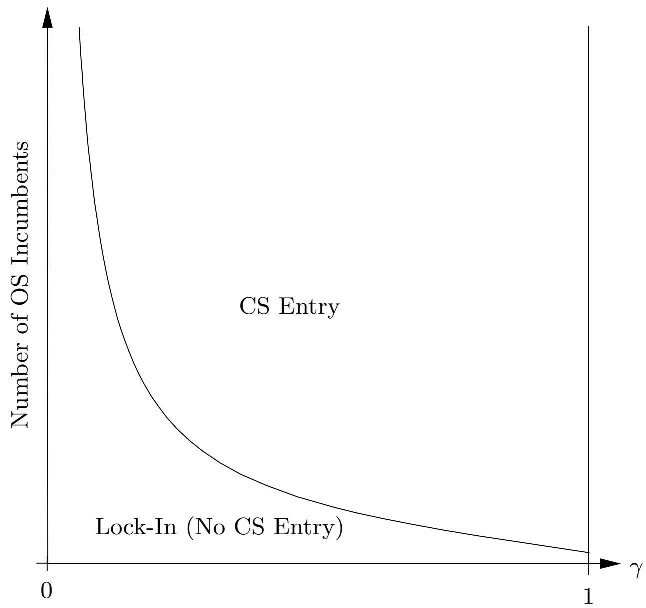
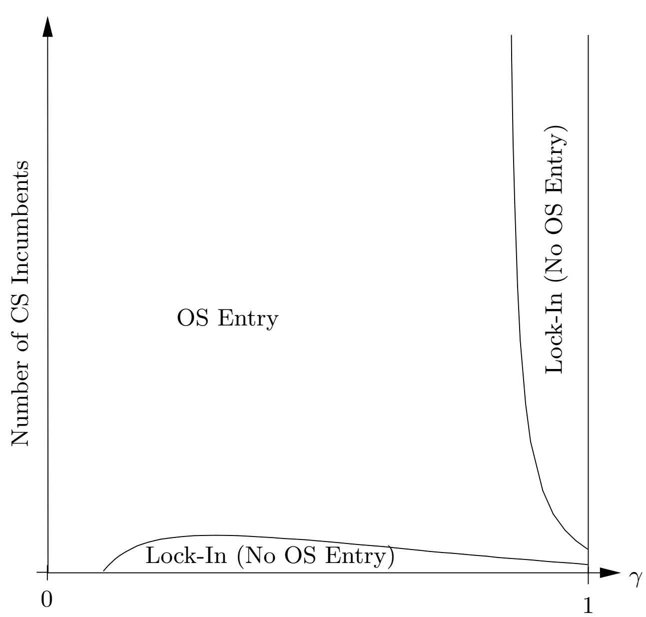


Figure 8: CS Lock-In



locked into Pure states where no further entry occurs. This can give incumbents a strategic reason to choose CS over OS in Stage Zero. Incumbents will strategically choose CS over OS whenever doing so will (a) block OS entrants, (b) block additional CS entrants, and (c) produce greater profits than incumbents would earn by choosing OS.[26] It can be shown that there are indeed constellations of parameters where incumbents can deliberately choose CS over OS in order to lock in a Pure-CS industry and secure positive oligopoly profits.

Proposition 11. *There exist cases where pure industries become "locked in" such that CS firms are unable to profitably enter Pure-OS industries and OS firms are unable to enter Pure-CS industries. There exist cases where first-mover incumbents strategically adopt CS in order to lock in a Pure-CS market where they receive positive oligopoly profits.*

Proof. 1.) For Pure-OS, the zero profit condition for the z OS incumbents, $\pi^{\text{os}}_{n=z} = 0$, yields Stage Two costs $C_{n=z} = p^{\text{os}}q^{\text{os}} - c^{\text{os}}_i$. CS entry only occurs if the entrant's profits $\pi^{\text{cs}}_{n=z+1} = p^{\text{cs}}q^{\text{cs}} - c^{\text{cs}} - C_{n=z}$ are greater than zero. Hence $\pi^{\text{cs}}_{n=z+1} = 0$ yields the boundary for CS entry. Setting for example $\phi = 2$ proves the existence of lock ins as depicted in Figure 7.

2.) For Pure-CS, the zero profit condition for the r CS incumbents, $\pi^{\text{cs}}_{n=r} = 0$, yields the corresponding Stage Two costs $C_{n=r} = p^{\text{cs}}q^{\text{cs}} - c^{\text{cs}}$. OS-entry occurs only if each entrant's profits $\pi^{\text{os}}_{n=r+2} = p^{\text{os}}q^{\text{os}} - c^{\text{os}} - C_{n=r}$ are greater than zero. Solving $\pi^{\text{os}}_{n=r+2} = 0$ for γ yields the boundary for OS entry. Setting $\phi = 2$ delivers a result as depicted in Figure 8. Higher ϕ yield larger low n/γ lock in region, while the high γ lock in region shrinks. □

Proposition 12. *There exist cases where first-mover incumbents strategically adopt CS in order to lock in a Pure-CS market where they receive positive oligopoly profits.*

Proof. This statement was proofed by setting $\phi = 2$ and then numerically finding cases. For example $n = 4$, $\gamma = 0.3$, $C = 0.1055$ yields the above stated result. □

[26] Profits are clearly higher for any given number of firms in Pure-OS markets compared to Pure-CS markets. However, this invites further entry by OS or CS firms and can drive profits below the Pure-CS case.

6. Government Intervention

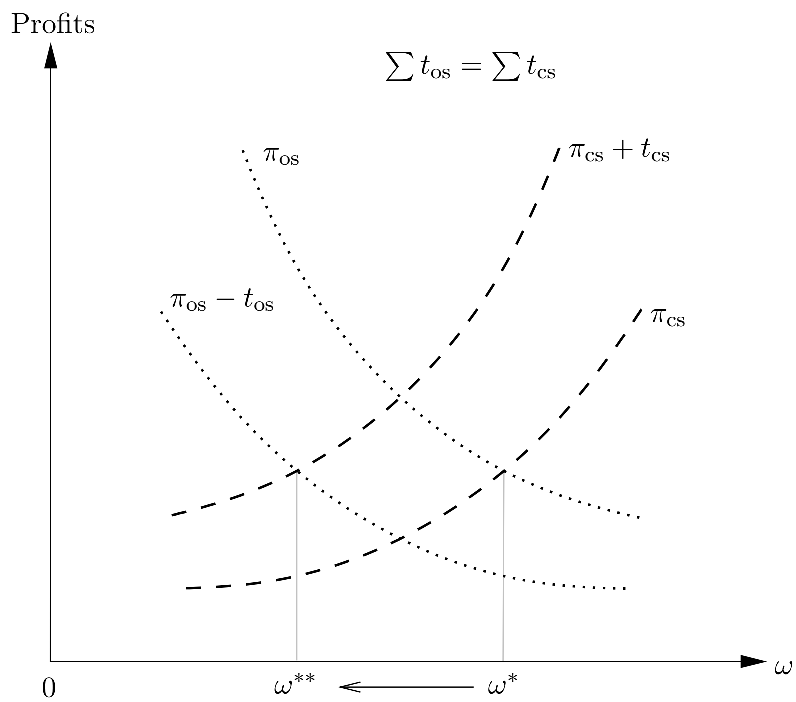
In recent years governments have begun experimenting with various pro-OS measures including procurement preferences, tax breaks, and grants (Lerner and Tirole, 2005; CSIS, 2008). So far, the most ambitious real world examples seem to be found in Singapore (tax breaks for commercial Linux adopters), Hong Kong (subsidies for firms that use or adopt OS) and Israel (grants to start-up companies that develop and use OS) (CSIS, 2008). This section compares the most common actual and proposed pro-OS government interventions with our model. The resulting insights must, of course, be handled with care. Real policy decisions inevitably address complex and poorly-known fact patterns that no reasonable theory can capture. Even so, policy interventions should be based on clear, articulable rationales. Policies that deliberately depart from theory should bear a special burden of proof.

6.1. Tax Policy

Tax/subsidy interventions are readily analyzed in our model. To begin with, we have seen that the proportion of OS firms in Mixed-OS/CS industries is higher than welfare-optimization requires. This result immediately suggests that giving OS firms tax breaks will be counterproductive. We therefore consider the opposite scheme where government imposes a fixed, lump-sum tax on OS firms and uses the proceeds to give lump-sum tax breaks or subsidies to CS firms. Detailed calculation shows that these interventions do indeed reduce the equilibrium ω in our model. Figure 9 presents a stylized illustration of this process. Optimal taxation adjusts the market outcome so that it delivers the desired, welfare-optimal mix of OS and CS firms. Furthermore, our hypothetical revenue transfer is accomplished through lump-sum taxes and payments. This leaves firms' optimal decision, and hence both the desired OS firm proportion and its expected welfare, the same as they were before.

There are, of course, practical objections to such a transfer. Most obviously, governments are not able to estimate the welfare optimal OS firm Proportion $\hat{\omega}$ and/or the required tax rates with any degree of precision. For this reason, any ambitious transfer scheme runs the risk of over-taxing OS firms and over-subsidizing their CS competitors. Even so, our objections to pro-OS fiscal measures stand. At a minimum, proponents should bear the burden of explaining why quality competition and de facto cartel effects can be safely ignored.

Figure 9: Lump-Sum Tax on OS Firms and Tax-Breaks for CS Firms



Policy Implication 1. *There is no theoretical support for tax breaks for OS firms. The first-best solution in the model is to tax OS firms and grant tax breaks to CS firms.*

6.2. Government "Provision" of OS Software

Many countries have established institutes, collaborations, grants, and partnerships to fund OS software development.[27] Additionally, government research grants often require academia and industry to prepare detailed dissemination plans when software is produced. OS is by far the easiest way to meet these obligations.[28] In principle government could also pay contractors directly to write OS code. All these measures would produce more OS than the market would otherwise supply. But do they improve welfare?

[27]Examples include China, Finland, Japan, South Korea, France, India, Slovakia, Spain, Thailand, Venezuela, and Vietnam. (CSIS, 2008).

[28]More formally, the United Kingdom has adopted a "default position" that government-funded software should be released under OS licenses.

The analysis is simplest for pure-CS markets where incumbents can otherwise block the entry of OS firms. Here, government-supplied OS reduces entry costs so that OS firms can overcome lock in.

The case for government provision of OS in mixed industries is more ambiguous because it has two implications. The *first* effect deals with firms' incentives to develop and to use OS code. Government-supported OS simultaneously (a) makes OS firms more profitable so that ω^* increases, and (b) crowds out OS firms' incentives to invest so that the welfare-optimal $\hat{\omega}$ falls. The net result is to make the already-large mismatch between the actual and desired mix of firms even worse. Indeed, very large government OS investments can drive CS firms out of the market entirely. The *second* effect of government-provided OS code deals with the cost-sharing benefit of OS. Suppose that government only cares about achieving the correct level of production whether or not private firms are involved.[29] Then the new, government-supplied code writers can be thought of as an additional OS firm that chooses output based on government fiat instead of profit-maximization. This lets policymakers select any desired level of quality while still obtaining significant code production from private OS firms.[30] Once again, however, government may not be able to estimate how much OS software to fund with any degree of precision.

Policy Implication 2. *Government-provided OS widens the gap between the welfare optimal and the stable proportion of OS firm and crowds out private OS investment. Since OS avoids duplication of costs and the crowding out is only partial, government provision can increase total OS and improve welfare.*

6.3. Government Procurement Preferences

Governments purchase large amounts of software. These purchases can potentially be used to promote OS over CS and vice versa. At least sixteen countries have considered mandatory polices that would require government agencies and/or state-owned companies to purchase OS solutions whenever possible.[31] Softer versions of these proposals speak of "preferences" for OS

[29]The argument is most easily seen where government contractors have the same cost structure as private OS firms. This very debatable assumption is not essential.

[30]Llanes and de Elejalde (2009) similarly predict that government investment in OS increases the amount of OS software while encouraging more OS firms to enter the market.

[31]Argentina, Belgium, Brazil, Bulgaria, Chile, Colombia, Israel, Italy, Netherlands, Ukraine, Finland, Portugal, Peru, and Venezuela.

when its performance is comparable to CS. To date, at least ten national governments have adopted some version of these proposals[32] along with many state and local governments (CSIS, 2008). High government adoption rates for OS in France and other countries suggest that unofficial preferences are also important.

Once again, the analysis is simplest for Pure-CS industries that would not otherwise evolve into welfare-improving Mixed states. Here, procurement preferences (and even requirements) can be a powerful tool for promoting OS entry that would not otherwise occur.

Analyzing the impact of procurement preferences on mixed industries is more subtle. Assume for the sake of definiteness that government ordinarily procures D bundles. In the benchmark (zero preferences) case government simply expands market demand and thus purchases one bundle from each firm so that each firm's demand is shifted by $d = \frac{D}{n}$. By comparison, adding government OS preferences means that each OS firm sees an extra demand of $d = \frac{D}{z}$ while each CS firm sees $d = 0$. This, however, increases the proportion of OS firms in equilibrium and widens the gap between actual and desired ω. However, unlike the case of government-provided OS, government now fails to add any new resources. This implies that the new $\omega^{\star\prime}$ offers less welfare than the $\omega^\star$ in our benchmark case. Conversely, pro-CS preferences[33] yield the opposite result and increase welfare.

Policy Implication 3. *Government procurement preferences for OS software increase the gap between the welfare optimal ($\hat{\omega}$) and the stable proportion ω^* and reduce total welfare.*

7. Testing the Model

We have presented a model which can be used wherever firms fund OS to improve the quality of bundled goods, services, and software. This generic quality is deliberate. Most proposed OS interventions (e.g. tax policy) require global rules. These can only be justified based on insights that cut across individual cases.

[32] Australia, Belgium, Brazil, China, Malaysia, the Netherlands, Peru, South Africa, Spain, and Venezuela. Conversely, the UK, Canada, Germany and Slovenia choose between OS and CS solely on technical merits.

[33] We assume that CS firms receive extra demand $d = \frac{D}{r}$ and OS firms have $d = 0$.

The only real question, then, is how well our model approximates reality. This section identifies eight empirical questions that can be used to test the model. For now, the literature is still in its infancy. We can, however, compare the model against the handful of studies that already exist along with less formal sources. While obviously preliminary, the results are encouraging.

Test 1: The Complements Assumption. Our model assumes that commercial firms pay programmers to write OS software which is later bundled with complementary goods and services. This hypothesis can be tested by tracing code deposits back through individual programmers to the firms that employ them. To date, we have located just one suitable database. The IBM-led Eclipse project, characterized as "the most vibrant" commercial OS project (West and Gallagher, 2006). builds so-called Integrated Development Environments ("IDEs") that are used to automate software production. Despite significant gaps in Eclipse's records, we have been able to trace roughly ninety percent of all deposits back to identifiable corporate sponsors. Strikingly, all of these firms produce complements whose quality depends on Eclipse software. These include proprietary software products that "plug into" Eclipse, consulting services for Eclipse users, Eclipse-based software for managing large data problems (e.g. train schedules), and custom programming services developed using Eclipse tools (Maurer, 2012).[34] We do not, of course, claim that our complements model applies to every commercial OS project without exception. As noted in Section 2.2, the limited available evidence suggests that counterexamples account for relatively little economic activity.

Test 2: Average Firm Size (Pt. 1). The model predicts that OS firms in mixed OS/CS industries will be smaller—i.e., possess less market share per firm—than their CS rivals (Engelhardt, 2010, see also Section 5.1). A rigorous test will require market share estimates across a representative sample of OS/CS markets. In the meantime, the academic survey literature is suggestive. For example, Harison and Koski (2010)'s survey of Finnish software firms finds that OS firms systematically report smaller turnover than CS

[34]Data for other high-profile OS collaborations, though less complete, is similarly suggestive. For example, the Google-led Open Handset Alliance creates operating system for mobile devices like cell phones. While there seems to be no public record of code deposits, we do know the names of the project's 83 corporate sponsors. Once again, they consist entirely of firms that manufacture cell phones, microchips, telecom software and other complements.

firms. Similarly, Fritsch and von Engelhardt (2010) find that German OS ICT firms report less start-up capital and staff than their CS counterparts.

Test 3: Average Firm Size (Pt.2). Our model predicts that the size gap between OS and CS firms will be largest where OS collaboration members produce close substitutes. As with Test 2, researchers can test this model by compiling detailed market share data.

Test 4: OS Market Share (Pt. 1). Our model predicts that commercial OS collaborations will produce more software and enjoy more market share where each member serves a different niche market. Testing this prediction will require econometric cross-elasticity studies. In the meantime, we note that Harison and Koski (2010) find that Finnish OS firms offer more diverse services than their CS counterparts. This pattern is commonly found in industries where firms specialize in finding and serving niche markets. Detailed case studies would go some distance to confirming this conjecture.

Test 5: OS Market Share (Pt. 2). Throughout this paper we have fixed the quality of the complement. A more general version allows for different values (see Footnote 17 and Engelhardt, 2010). This model predicts that OS products should have more market share when the bundle's overall quality depends sensitively on the complementary product. One natural test would be to compare data for different types of software-driven hardware. In the meantime, it is instructive to compare the reported market shares for desktop operating systems, embedded Linux, and IDE software. In the desktop case, the power of modern computer chips is almost never a limiting factor. For this reason, we expect quality differences to depend almost entirely on software. This helps explain why open desktop software serves just one percent of the market (NetApplications 2012). Significantly, Linux's market share is larger for enterprise desktop where OS firms sell consulting and technical support. At the other extreme, the value of IDEs depends sensitively on programmer skill. This helps explain why IDEs currently enjoy a 97% market share (ZeroTurnaround, 2010). Finally, manufactured objects (e.g. refrigerators, airplanes) derive quality both from physical design and software. As expected, OS products account for roughly half of this market (Linux Devices.com, 2007).

Test 6: OS Market Share (Pt. 3). Our model predicts that ω and hence OS market share is greater where the marginal cost of software production is steeply increasing (see Appendix G). In principle, this can be tested by comparing OS against cost data for distinct project types, for example operating systems vs. database software. For now, however, attempts to estimate OS cost are still in their infancy (Dolado, 2001; Asundi, 2005; Koch, 2004).

Test 7: OS Quality. Our model predicts that OS firms will usually offer lower quality bundles than their CS competitors. However, existing measures of software quality are complex, controversial, and sometimes indeterminate. The question may be easier to research for software-driven consumer goods. Automakers' increasing use of open source software suggests a promising test case (BearingPoint, 2012).

Test 8: Interaction With Other OS Communities. In cases where non-commercial communities provide large amount of OS code, the model predicts increases in both (a) the total number of OS firms, and (b) the market share of products that contain OS software. In principle, the effect should be visible in projects where the fraction of non-commercial donors changes over time. As we have seen, however, such donors are hard to identify; furthermore, their participation will often be correlated with other secular changes over the life of the project. For this reason, researchers may find it simpler to study what happens when large non-commercial entities (e.g. computer science departments, national labs) enter or leave a particular project.

8. Discussion

We have presented a general model that allows us to compare equilibrium OS:CS firm ratios against the target ratios that would be needed to maximize welfare. This section explores the extent to which extended models would qualitatively change our results.

8.1. Basic Analysis

Our model OS output reflects a delicate balance between (a) lower per-firm costs through shared development, (b) reduced appropriability and hence smaller investment incentives in case of strong quantity competition, and (c) a cartel effect that suppresses quality competition among OS firms. The first two effects are fairly straightforward and have previously been noted in, for example, Llanes and de Elejalde (2009), and Henkel (2006a).

To the best of our knowledge, however, our third ("quality-cartel") factor has never been noticed before. This is puzzling because Llanes and de Elejalde (2009)'s model is somewhat similar to ours. We conjecture that quality cartels do not appear in their model because they require firms to set quality and prices in a single, simultaneous decision. Introducing a single simultaneous decision similarly suppresses quality competition (and therefore quality-cartel effects) in our model.[35] We therefore expect a similar "quality-cartel" effect to appear in an extended Llanes and de Elejalde model—and, indeed, most generic models that require firms to make their quality and price decisions sequentially. Such models are also more realistic given the lags between firms' software development plans and pricing/output decisions.

8.2. Technology Assumptions

The quality of our bundles is determined by a single technology (software) which is (a) indivisible, (b) can be developed jointly, (c) can be shared, and (d) confers an identical quality "boost" on every product. In principle, all of these assumptions can be relaxed.

Our first assumption that all quality comes from a single indivisible technology ("software") is more general than it looks since "software" can trivially be relabeled (a) to include other technologies (e.g. hardware) and (b) to exclude any technology (including software) that is only usable by the author. Assuming that software is indivisible, however, does exclude situations where quality depends on two or more separate and distinct technologies. One natural speculation is to ask what happens when firms can also invest in a second technology whose benefits are primarily limited to their own Stage Two product. Intuitively, we would expect this additional quality investment to drain resources from Stage One R&D leading to fewer OS firms, less cost-sharing, and less development of shared software. This is more or less what happens when Llanes and de Elejalde (2009) allow firms in their model to invest in a second technology focused narrowly on their products. The existence of a second, severable technology also facilitates strategic behaviors in which firms keep at least one technology closed as a barrier to entry (Schmidtke, 2006).

[35] We analyzed a one-stage version of our model where firms choose the profit-maximizing (price, quality) pair. In the Pure-CS case quality competition is so weak that output continues to decline even at high γ. In the Pure-OS case, our results approximate the results for a formal OS quality-cartel described in Appendix C

Relaxing the second assumption that firms can develop code jointly is strongly model-dependent and should probably await convincing evidence that such failings actually exist. In the meantime, we note that Henkel (2006a) has explored a model in which joint development is impractical at the level of individual OS modules so that each project is effectively controlled by one (and only one) company. Henkel argues that firms self-select toward developing whichever modules they value most and that this biases total OS investment upward. More generally, one can also imagine models in which OS joint development is possible but inefficient or imperfectly monitored. This could happen, for example, if participants adopted mixed strategies that encouraged them to strategically withhold effort from the collaboration in hopes that some other member would do the work (Johnson, 2002)

The third assumption that firms can share OS might be relaxed if, for example, substantial "tacit knowledge" was needed to use completed software. We have explored this scenario by adding a "spillover parameter" $\sigma \in]0,1]$, such that $\alpha_i = 1 + x_i^{\text{os}} + \sigma X_{-i}^{\text{os}}$. We find that our OS results gradually converge to CS where spillovers are small.

Finally, completed software may not boost all firms' products quality equally. Naively, we would expect the presence of firms that gain relatively little from OS to produce free-rider effects. Relatedly, firms' willingness to invest in OS could depend on the size of their respective Stage Two markets. It would be natural to investigate this by allowing different qualities of the Stage Two products in our model. For now, it is probably safe to say that the answer will sensitively depend on how many separate technologies exist and the distribution of preferences among firms. Absent detailed empirical guidance, it will be hard to know which models to investigate.

8.3. Demand Side Assumptions

Our model assumes that consumers choose between products based on quantity supplied, substitutability, and a one-dimensional quality parameter that reflects the amount of software produced. We recognize, however, that consumers may have idiosyncratic preferences for particular products. Naively, we expect strong consumer preferences to reduce the payoffs from quality improvements leading to lower code production. To the best of our knowledge, Llanes and de Elejalde (2009) are the only authors who have investigated firms' decisions to invest in quality using a Hotelling model that includes idiosyncratic demand and allows for $n > 2$ firms. While their results are broadly similar to ours, Pure-OS industries are indeed much more common in their model. We conjecture that idiosyncratic consumer preferences reduce the importance of appropriability so that shared OS software production becomes more lucrative.

Similarly, the degree of substitutability (γ) is exogenous in our model. Over time, however, one might expect firms to design new products strategically so that γ becomes endogenous. This could be accomplished by, for example, linking our two appropriability variables n and γ. Alternatively, one might think that shared code would make OS products resemble each other more closely (high γ) than they do CS products (low γ). Llanes and de Elejalde (2009) explore this possibility by introducing different substitutability parameters for bundles that contain OS compared to bundles that contain CS. Not surprisingly, they find that increased substitutability leads to greater competition among OS firms which, in turn, makes CS firms larger and more profitable. Endogenizing substitutability in our model would probably produce similar results.

Finally, we have limited our analysis to the case where products are substitutes. However, not all products compete and some are complements. As Schmidtke (2006) points out, OS provides a natural way for firms to encourage the production of complements that will increase their own product sales. Extending the current model to include this case would probably mitigate the free rider but not the cartel effect. Furthermore, the number of such complementary products—and hence the importance of Schmidtke's observation—remains unclear.

8.4. Non-Commercial OS Incentives

We started this paper by remarking that commercial incentives to develop OS has become increasingly important in recent years However, volunteer labor remains important for many OS collaborations and dominates some. In principle, some of this voluntarism may reflect the desire for future wages and could be endogenized in our model as a kind of prize. In general, however, many motives (reputation, altruism, fun) fall outside neoclassical economics. We refer to OS code developed for these reasons as 'traditional OS' in what follows.

The effects of traditional OS are similar to those already considered for government provided OS. Thus, increased traditional OS (a) increases the total supply of OS, (b) increases the total number of OS firms, and (c) reduces the total amount of OS produced for commercial reasons. The main difference is that non-commercial OS potentially increases welfare faster than government OS. The reason is that programmers who gain psychic benefits from voluntarily supplying non-commercial OS have already been compensated because of their intrinsic motives (Lakhani and Wolf, 2005; Hertel et al., 2003; Ghosh et al., 2002b). By definition, such activities have no opportunity costs and thus increase welfare even more (Pasche and Engelhardt, 2004).

9. Conclusions and Outlook

Today's OS software is increasingly dominated by business strategies in which firms make proprietary products whose quality depends on a shared OS code base. We have presented a generic oligopoly model based on simple, realistic assumptions about the costs of developing OS and CS code. We find that Pure-OS industries are welfare-superior to Pure-CS states so long as quantity competition is modest so that appropriability is high. Otherwise, Pure-CS industries are welfare-superior except for a few industries where strong quality competition leads to over-investment by CS firms. Finally, mixed industries are potentially superior to both Pure-OS and Pure-CS industries. Welfare is maximized when the presence of CS firms introduces quality competition that drives OS firms to produce large amounts of code. Ironically, then, OS is only able to realize the full benefits of cost-sharing when CS firms are present.

Unfortunately, our analysis also shows that ordinary markets do not produce nearly enough CS firms to maximize welfare. Additionally, industries can be locked into pure markets so that they never transition to welfare-improving mixed markets. Finally, incumbents can sometimes deliberately block entry by choosing CS.

This leads to several policy implications. Our model provides no support for giving OS firms tax breaks. To the contrary: its most elegant proposal is to increase welfare by taxing OS firms and using the proceeds to subsidize CS firms. Similarly, procurement preferences that concentrate government spending on OS bundles should be viewed with suspicion. Such schemes invariably decrease welfare and make the mismatch between the actual and welfare-optimal number of OS firms even worse. The model does however provide support for policy measures that directly boost the amount of supplied OS code. This would increase welfare over the market outcome despite widening the gap between the actual and welfare optimal proportion of OS firms.

Appendix A. Bertrand Competition

This section presents a general version of the model that includes both Cournot and Bertrand competition. We follow Häckner (2000)'s solution of Cournot and Bertrand competition in horizontally and vertically differentiated oligopolies. This yields the following payoff function for firm i in Stage One.

$$\max_{x_i} \left(\alpha_i A - V \sum_{j \neq i} \alpha_j \right)^2 H^{-1} - c_i \left(x_i, \cdot \right)$$

where (a) in the case of *Cournot* competition $A = 1 + (n-1)\theta$, $V = \theta$ and $H = h^2$, with $\theta = \frac{\gamma}{(2-\gamma)}$ and $h^2 = 2 + \gamma(n-1)$; and (b) in the case of *Bertrand* competition $A = \gamma^2 \left[(n-2)^2 - (n-1) \right] + 3\,(n-2)\,\gamma + 2$, $V = \gamma\,((n-2)\,\gamma + 1)$ and $H = (\gamma\,(n-3) + 2)^2\,(2 + \gamma\,(2\,n-3))^2\,\frac{(1-\gamma)(\gamma\,(n-1)+1)}{(n-2)\gamma+1}$. Note that $A > 0$, $U > 0$, and $V > 0$ for all $n \geq 2$, $n \in \mathbb{N}$ regardless of whether Cournot or Bertrand is applied. Note also that $\gamma \in [0 \ldots 1[$ for Bertrand competition because of the general demand function used in Häckner (2000).

In case of restricted licenses we get the following reaction functions for an OS-firm and CS-firm respectively:

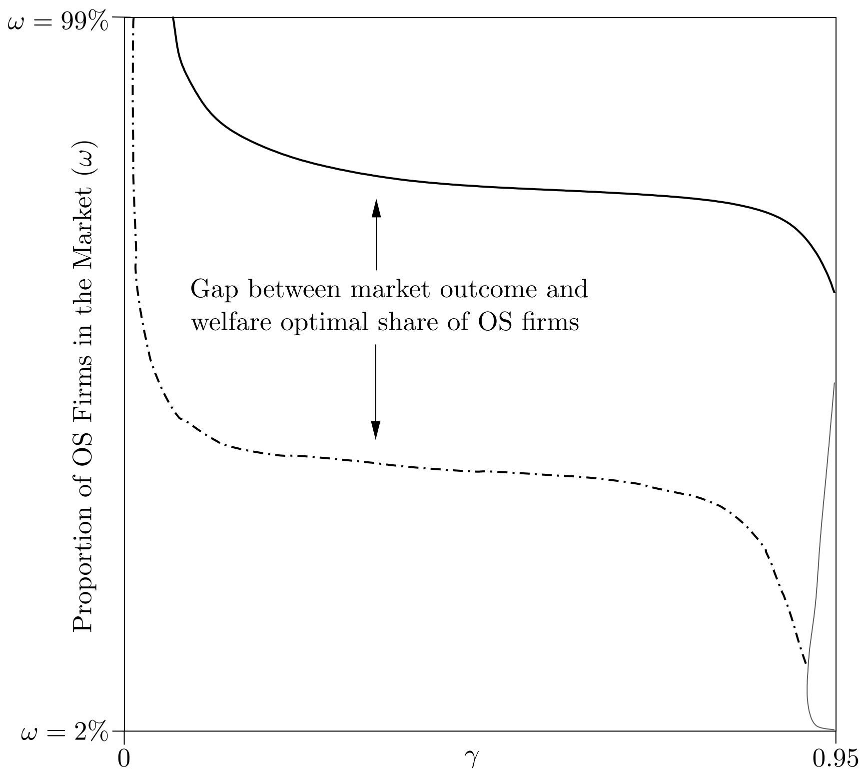
$$R^{cs}_{i\in R} = \beta \frac{A\,(A - V\,(n-1))}{\frac{1}{2}\phi H - A^2} - zX^{os} \frac{AV}{\frac{1}{2}\phi H - A^2} - \sum_{j \neq i} x^{os}_j \frac{AV}{\frac{1}{2}\phi H - A^2}, \quad \text{(A.1)}$$

$$\begin{aligned} R^{os}_{i\in Z} = \beta &\frac{(A - V\,(n-1))\,(A - V\,(z-1))}{\frac{1}{2}\phi H - (A - V\,(z-1))^2} - \sum_{j \neq i} x^{os}_j \frac{V\,(A - V\,(z-1))}{\frac{1}{2}\phi H - (A - V\,(z-1))^2} \\ &- \frac{\left(\frac{1}{4}\phi - (A - V\,(z-1))^2\right) X^{\mathrm{os}}_{-i}}{\frac{1}{2}\phi H - (A - V\,(z-1))^2} \end{aligned} \quad \text{(A.2)}$$

where z is the number of OS-firms and r is the number of CS-firms and $r + z = n$. Now, $A > 0$, $H > 0$ and $V > 0$ for all $n \geq 2$, $n \in \mathbb{N}$ and because of the SOC is $\frac{1}{2}\phi H - (A - V\,(z-1))^2 > 0$ and $\frac{1}{2}\phi H - A^2 > 0$.

We now turn to a mixed market equilibrium with free entry and exit. Taking the SOC and no-cut-off boundary into account, it is again possible to compare the stable $\omega^\star$ derived from the zero-profit condition against the welfare optimal $\hat{\omega}$. Figure A.10 depicts a representative example for the case of Bertrand competition (taking into account that $\gamma < 1$). The reader can confirm the similarity to the Cournot case depicted in Figure 6

Figure A.10: Welfare-Comparison of Pure vs. Mixed Cases: Bertrand



This similarity between Bertrand and Cournot results hold in general. This is confirmed by various numerical examples available upon request from the authors. The general form of the reaction functions above shows that assuming Bertrand or Cournot does not change the strategic interactions that drive our results. For example, the incentives to invest in quality seems to be higher in the Bertrand case. However, these different incentives affect both OS and CS firms. This explains why the model results of ω^* versus $\hat{\omega}$ are similar.

Appendix B. Impact of Asymmetry of Total Costs between OS and CS

Some commentators have argued that OS is inherently cheaper to develop than CS software (Raymond, 1998). We now discuss how such a cost asymmetry would affect our model.

A more general formulation of the OS cost function is

$$c_i(x_i^{\text{os}}) = \frac{1}{2}\phi x_i^{\text{os}}\left(x_i^{\text{os}} + \lambda \cdot X_{-i}^{\text{os}}\right), \quad \lambda \geq 0 \tag{B.1}$$

where $\lambda = 1$ for the case of cost symmetry, and $\lambda \lesseqgtr 1$ otherwise. This leads to a slightly different solution

$$x^{\text{os}} = \frac{(1 + r\theta)(\beta - \theta r x^{\text{cs}})}{\frac{1}{4}\phi h^2 (2 + \lambda (z - 1)) - z (1 + r\theta)^2} \tag{B.2}$$

Readers can easily confirm that lower values of λ lead to higher values of x^{os}.

When OS has a cost advantage ($\lambda < 1$) it becomes more attractive so that the SOC value of ϕ is reduced and there are more OS firms in equilibrium (when CS has a cost advantage the opposite holds). The $\lambda = 0$ case leads to a strong superadditive OS cost function and to Pure-OS markets only.

Appendix C. OS versus a 'Real' OS Quality-Cartel

We compare the outcome of individual investment decisions by OS firms against a cartel in which firms maximize joint profits.

Appendix C.1. OS-Firms, No Formal Cartel

In case of OS with competition, each OS firm $i = 1 \ldots n$ maximizes its profits. Firm i's reaction function is given by (5). By symmetry this same reaction function governs all $i = 1 \ldots n$ firms. This determines the equilibrium value of $X^{\text{os}} = \sum_i x_i^{\text{os}} = n \cdot x^{\text{os}}$ as given by (7):

$$X^{\text{os}*} = \frac{n}{\frac{1}{4}\phi h^2 (1 + n) - n}$$

Appendix C.2. A Formal OS Quality-Cartel

Consider a 'real' cartel in which firms maximize joint profits $\Pi = \sum_i \pi_i^{\text{os}}$. Then profit maximization requires:

$$\max_{X^{\text{os}}} \Pi = \sum_i \pi_i^{\text{os}} = n\frac{(1 + X^{\text{os}})^2}{h^2} - \frac{1}{2}\phi X^{\text{os}2}$$

Setting the first derivative equal to zero yields the cartel-output

$$X^{\text{os, cartel}} = \frac{n}{\frac{1}{2}\phi\, h^2 - n} \; \forall \phi > \frac{2n}{h^2}$$

Appendix C.3. Comparing Outcomes

For all $\phi > \frac{2n}{h^2}$ (the second order condition of the cartel), the cartel produces more code than individual firms do:[36] $X^{\text{os, cartel}} > X^{\text{os}*}$. Furthermore firms also earn higher profits, i.e. $\pi_i^{\text{os, cartel}} > \pi_i^{\text{os}*}$. Finally, $W^{\text{os, cartel}} < W^{\text{os}*}$.

Appendix D. Welfare

Our analysis is based on Hsu and Wang (2005) who provide a general welfare analysis for differentiated oligopolies. Applied to our model, this yields the *consumer surplus* A as follows:

$$A = \frac{1}{2}\left((1-\gamma)\left(zq_{i\in Z}^2 + rq_{i\in R}^2\right) + \gamma\left(zq_{i\in Z} + rq_{i\in R}\right)^2\right)$$

with

$$q_{i\in Z} = \frac{\left(1 + zx^{\text{os}*} - r\theta\left(x^{\text{cs}*} - zx^{\text{os}*}\right)\right)}{h},$$

$$q_{i\in R} = \frac{\left(1 + x^{\text{cs}*} - z\theta\left(zx^{\text{os}*} - x^{\text{cs}*}\right)\right)}{h}.$$

The *producer surplus* B is given by

$$B = z\cdot\pi_{i\in Z} + r\cdot\pi_{i\in R},$$

with profits

$$\pi_{i\in Z} = \frac{\left(1 + zx^{\text{os}*} - r\theta\left(x^{\text{cs}*} - zx^{\text{os}*}\right)\right)^2}{h^2} - \frac{1}{2}\phi z x^{\text{os}*2},$$

$$\pi_{i\in R} = \frac{\left(1 + x^{\text{cs}*} - z\theta\left(zx^{\text{os}*} - x^{\text{cs}*}\right)\right)^2}{h^2} - \frac{1}{2}\phi x^{\text{cs}*2}.$$

Appendix D.1. Pure Cases

In case of Pure-OS ($n = z$) we obtain

$$W_n^{\text{os}} = A_{n=z} + B_{n=z} = (1+h)\,\frac{n}{2}\frac{\left(1 + nx^{\text{os}*}\right)^2}{h^2} - \frac{1}{2}\phi\left(nx^{\text{os}*}\right)^2.$$

[36] If the second-order condition for cartels is not satisfied, i.e. $\phi < \frac{2n}{h^2}$, the cartel produces software up to the cut-off.

In case of Pure-CS ($n = r$) we obtain

$$W_n^{\text{cs}} = A_{n=r} + B_{n=r} = (1+h)\,\frac{n}{2}\frac{\left(1+x^{\text{cs}*}\right)^2}{h^2} - n\frac{1}{2}\phi x^{\text{cs}*2}.$$

We can now calculate the *difference in welfare* ($\Delta W_n = W_n^{\text{os}} - W_n^{\text{cs}}$) for a given n:

$$\Delta W_n = W_n^{\text{os}} - W_n^{\text{cs}} = \frac{n}{2}(1+h)\,\frac{\left(1+x^{\text{os}*}n\right)^2 - \left(1+x^{\text{cs}*}\right)^2}{h^2} - \frac{n}{2}\phi\left(nx^{\text{os}*2} - x^{\text{cs}*2}\right).$$

with $nx^{\text{os}*} = X^{\text{os}*}$ is given by (7), and $x^{\text{cs}*}$ is given by (4).

Appendix D.2. Mixed Cases

The simultaneous solution of (8) and (9) is given by

$$x^{\text{cs}*} = \frac{(1+(n-1)\,\theta)\left(z^2\theta\,(1+\theta r)-\chi\right)\beta}{(1+\theta r)\,\theta^2 r\,(1+(n-1)\,\theta)\,z^2 - \psi\chi},$$
$$x^{\text{os}*} = \frac{(1+\theta r)\left(\theta r\,(1+(n-1)\,\theta)-\psi\right)\beta}{(1+\theta r)\,\theta^2 r\,(1+(n-1)\,\theta)\,z^2 - \psi\chi},$$

where $\psi = \frac{1}{2}h^2\phi - (1+(n-1)\,\theta)(1+z\theta)$ and $\chi = \frac{1}{4}\phi h^2(1+z) - z(1+r\theta)^2$. Inserting these in to our quantity and profits functions (see above) lets us calculate total welfare according to the expression

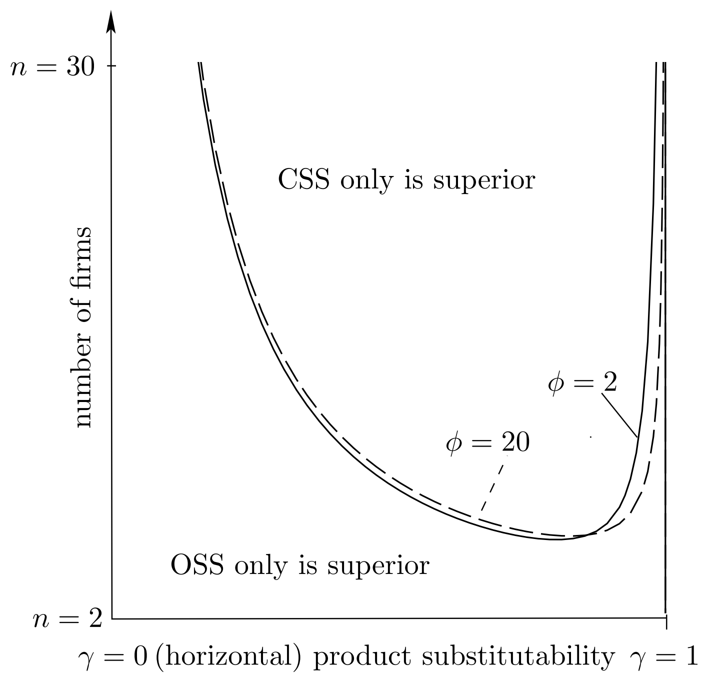
$$W = \frac{1}{2}\left((1-\gamma)\left(zq_{i\in Z}^2 + rq_{i\in R}^2\right) + \gamma\left(zq_{i\in Z} + rq_{i\in R}\right)^2\right) + z\cdot\pi_{i\in Z} + r\cdot\pi_{i\in R}.$$

Appendix E. Different ϕ and Welfare Comparison of Pure Cases

We show that adopting values of ϕ different from 2 does not substantially change our welfare comparison of Pure-OS and Pure-CS industries. Figure E.11 shows results for $\phi = 2$ (solid line) and $\phi = 20$ (dashed line).

The situation does not change much when ϕ increases by a factor of ten. The region where OS is superior expands slightly for low values of γ and contracts slightly near $\gamma = 1$. (Notice that we have exaggerated the differences by drawing the figure so that the affected regions are magnified.) The reason for this similarity is that changes in ϕ affect the cost function for Pure-OS and Pure-CS industries identically. Furthermore, the values of γ for which (a) $X^{os} = x^{cs}$ and (b) x^{cs} has its minimum (and begins to rise again) are given by (a) $\gamma = \frac{2}{(n+2)}$ and (b) $\gamma = \frac{\left(n-5+\sqrt{9+n(n-2)}\right)}{2(n-2)}$, both independent of ϕ.

Figure E.11: Welfare Superiority of $\phi = 2$ vs. $\phi = 20$



Appendix F. Welfare of Pure vs. Mixed Cases in Concentrated Industries

For small industries (small n) one has to take into account that the number of firms is an integer. Figures F.12 and F.13 (page 51) show how our welfare analysis of Pure vs. Mixed cases changes for concentrated (small n) industries. While the case of $n = 30$ looks similar to the figure in the main text, in the case of $n = 5$ some regions have disappeared.

Appendix G. Stable vs. Welfare Optimal Proportion of OS Firms, and Various ϕ and n

We have performed various different numerical calculations to confirm our propositions. Table G.2 (page 52) presents selected results for different values of γ, ϕ and n. Further variations and/or the applied Maple code are available from the authors.

Figure F.12: Welfare-Comparison of Pure vs. Mixed Cases, $n = 30$

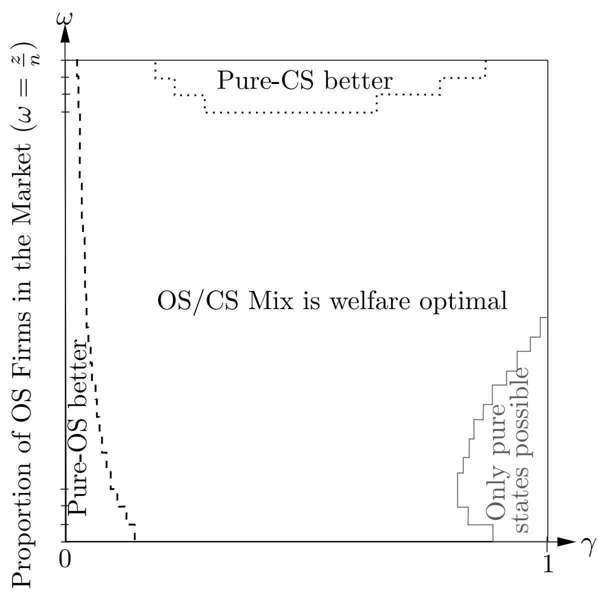


Figure F.13: Welfare-Comparison of Pure vs. Mixed Cases, $n = 5$

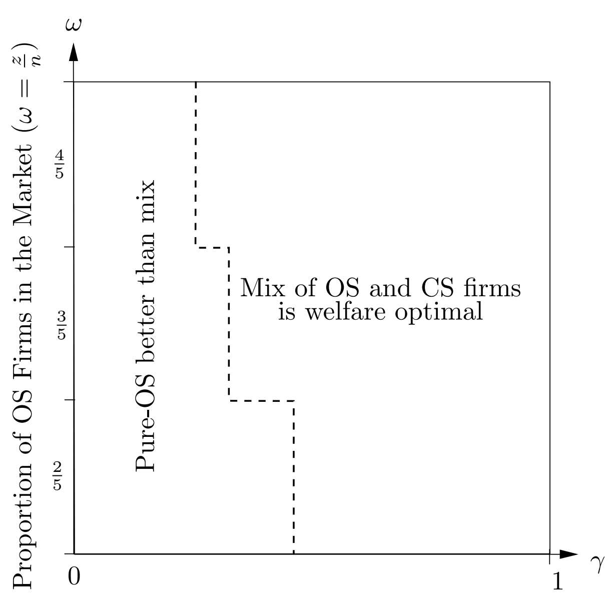
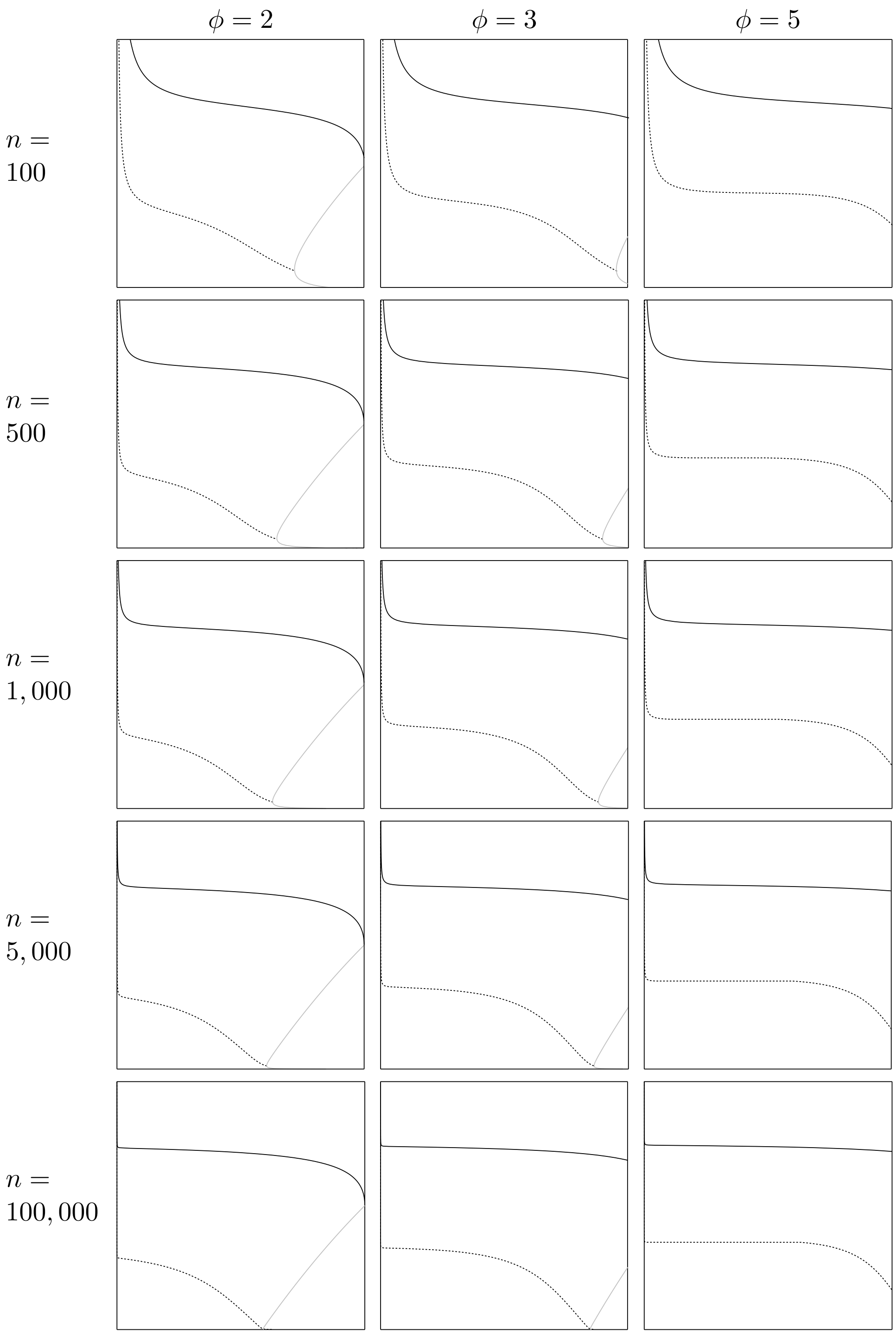


Table G.2: Stable vs. Welfare Optimal ω



Appendix H. OS-Entry into a Pure-CS Industry

We assume that two OS firms are potential entrants.[37]

Assume further that the two firms decide sequentially in Stage Zero. Then the first OS firm will not enter the market unless it knows that the second firm will also enter afterward. This means, by induction, that it will enter if and only if the second firm is assured of earning a profit. But this is exactly what the GPL license does. By making its own code GPL, the first entrant commits to the specific OS regime that allows the second entrant to earn a profit. We therefore conclude that there is no fundamental reason why Pure-CS states cannot transition to Mixed-OS/CS states so long as GPL-like commitment strategies exist.

Alternatively, assume that the potential OS entrants decide simultaneously. Suppose further that the two OS firms would earn a positive profit if both enter. But if only one OS firms enters, code-sharing cannot occur and the entrant will earn negative profits. Then the strategic problem is to ensure that both firms enter. This problem can be analyzed in terms of the following game where the payoffs have been normalized to 1, 0, and -1:

	YES	NO
YES	1,1	-1,0
NO	0,-1	0,0

This coordination game has two Nash-equilibria (YES, YES) and (NO, NO). Furthermore, this is a common interest game in which both two potential entrants would like to occupy the same (YES, YES) equilibrium. If players can signal which strategy they wish to play, we can assume that they will both arrive the common interest equilibrium. This can readily be done in our OS case if each OS entrant announces that its code will be subject to the GPL.

[37] Our argument does conceptually not depend on this assumption and holds equally for more OS-entrants.

Acknowledgement

We are grateful to Suzanne Scotchmer, Andreas Freytag, Maria Alessandra Rossi, the participants of the "Sixth bi-annual Conference on the Economics of Intellectual Property, Software and the Internet" (Toulouse, January 2011), the "Open Source, Innovation, and Entrepreneurship" workshop (Jena, 14 January 2010) and the "Augustin Cournot Doctoral Days" (Strasbourg, 1-3 April 2009) for their comments on this paper. Sebastian von Engelhardt gratefully acknowledges financial support from the Klaus Tschira Foundation. Stephen Maurer gratefully acknowledges financial support from the Alfred E. Sloan Foundation and the U.S. National Science Foundation (Grant No. SES 20082737). The authors are solely responsible for any errors and oversights.

References

Almeida, P., Kogut, B., 1999. Localization of knowledge and the mobility of engineers in regional networks. Management Science 45, 905–917.

Arora, A., Gambardella, A., 2010. Ideas for rent: an overview of markets for technology. Industrial and Corporate Change 19, 775–803.

Asundi, J., 2005. The need for effort estimation models for open source software projects, in: Proceedings of the Fifth Workshop on Open Source Software Engineering, ACM, New York, NY, USA. pp. 1–3.

Audretsch, D.B., Feldman, M.P., 1996. R&d spillovers and the geography of innovation and production. The American Economic Review 86, 630–640.

Baake, P., Wichmann, T., 2004. Open Source Software, Competition and Potential Entry. Berlecon Research Papers 5. Berlecon Research. Berlin.

Barnett, J.M., 2011. The host's dilemma: Strategic forfeiture in platform markets for informational goods. Harvard Law Review 124, 18611938.

BearingPoint, 2012. FOSS Management Study. Study. BearingPoint GmbH. `http://www.bearingpoint.com/en-other/7-5601/study-foss-management/`.

Bitzer, J., 2004. Commercial versus open source software: The role of product heterogeneity in competition. Economic Systems 28, 369 – 381.

Bresnahan, T., Gambardella, A., Saxenian, A., 2001. 'old economy' inputs for 'new economy' outcomes: Cluster formation in the new silicon valleys. Industrial and Corporate Change 10, 835–860.

Brooks, F.P., 1982. The Mythical Man-Month: Essays On Software Engineering. Addison-Wesley.

Buono, F.M., Sieverding, M.B., 2009. Government procurement of software: Provident policies for ensuring the greatest possible return on investment in troubled economic times. Bloomberg Law Reports.

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., 2011. Mixed source. Management Science 57, 1212–1230.

Choi, J.P., 1993. Cooperative R&D with product market competition. International Journal of Industrial Organization 11, 553 – 571.

Comino, S., Manenti, F.M., 2005. Government policies supporting open source software for the mass market. Review of Industrial Organization 26, 217–240.

Comino, S., Manenti, F.M., 2011. Dual licensing in open source software markets. Information Economics and Policy 23, 234 – 242.

Comino, S., Manenti, F.M., Rossi, A., 2011. Public intervention for free/open source software. Revue d'économie industrielle 136, 89–108.

CSIS, 2008. Government open source policies. Center for Strategic and International Studies. Available at `http://csis.org/`.

Daffara, C., 2007. Business models in FLOSS-based software companies. Workshop presentation at the 3rd Conference on Open Source Systems. `http://opensource.mit.edu/papers/OSSEMP07-daffara.pdf`.

Daffara, C., 2009. The small/medium enterprise guide to open source software. Guide in the context of the FLOSSMETRICS EU project. 4th ed.

Dahl, M., Pedersen, C., 2004. Knowledge flows through informal contacts in industrial clusters: myth or reality? Research Policy 33, 1673 – 1686.

Dahlander, L., 2007. Penguin in a new suit: A tale of how de novo entrants emerged to harness free and open source software communities. Industrial and Corporate Change 16, 913 – 943.

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.

Deshpande, A., Riehle, D., 2008. The total growth of open source, in: Russo, B., Damiani, E., Hissam, S., Lundell, B., Succi, G. (Eds.), Open Source Development, Communities and Quality. Springer. volume 275 of *IFIP International Federation for Information Processing*, pp. 197 – 209.

Dixit, A., 1979. A model of duopoly suggesting a theory of entry barriers. Bell Journal of Economics 10, 20–32.

Dolado, J.J., 2001. On the problem of the software cost function. Information & Software Technology 43, 61–72.

Engelhardt, S.v., 2010. Quality Competition or Quality Cooperation? License-Type and the Strategic Nature of Open Source vs. Closed Source Business Models. Jena Economic Research Papers 2010-034. Friedrich-Schiller-University Jena and Max-Planck-Institute of Economics.

Evans, D.S., Reddy, B., 2003. Government preferences for promoting open-source software: A solution in search of a problem. Michigan Telecommunications and Technology Law Review 9.

Fershtman, C., Gandal, N., 2004. The Determinants of Output Per Contributor in Open Source Projects: An Empirical Examination. CEPR Discussion Paper 4329. Centre for Economic Policy Research.

Fershtman, C., Gandal, N., 2011. Direct and indirect knowledge spillovers: the "social network" of open-source projects. The RAND Journal of Economics 42, 70–91.

Fink, M., 2002. Business and Economics of Linux and Open Source. Prentice Hall Professional Technical Reference.

Fritsch, M., von Engelhardt, S., 2010. Who Starts with Open Source? Institutional Choice of Start-Ups in the German ICT Sector. Jena Economic Research Papers 2010-049. Friedrich-Schiller-University Jena and Max-Planck-Institute of Economics.

Ghosh, R.A., 2006. Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU. Study. European Commission.

Ghosh, R.A., Glott, R., Krieger, B., Robles, G., 2002a. FLOSS Final Report, Part 3: Basics of Open Source Software Markets and Business Models. Technical Report. International Institute of Infonomics, University of Maastricht.

Ghosh, R.A., Glott, R., Krieger, B., Robles, G., 2002b. 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., Koski, H., 2010. Applying open innovation in business strategies: Evidence from finnish software firms. Research Policy 39, 351–359.

Harvey, C., 2010. 50 open source replacements for really expensive software. article on Datamation.com. `http://www.datamation.com/osrc/article.php/3898906/50-Open-Source-Replacements-f`

Henkel, J., 2006a. 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., 2006b. Selective revealing in open innovation processes: The case of embedded linux. Research Policy 35, 953 – 969.

Henkel, J., 2009. Champions of revealing–the role of open source developers in commercial firms. Industrial and Corporate Change 18, 435 – 471.

Henkel, J., Hippel, E.v., 2005. Welfare implications of user innovation. The Journal of Technology Transfer 30, 73–87.

Hertel, G., Niedner, S., Herrmann, S., 2003. Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. Research Policy 32, 1159 – 1177.

Hinloopen, J., 2003. R&D efficiency gains due to cooperation. Journal of Economics 80, 107–125.

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.

Kamien, M.I., Muller, E., Zang, I., 1992. Research joint ventures and R&D cartels. The American Economic Review 82, 1293–1306.

Koch, S., 2004. Profiling an open source project ecology and its programmers. Electronic Markets 14, 77–88.

Koski, H., 2005. OSS production and licensing strategies of software firms. Review of Economic Research on Copyright Issues 2, 111 – 125.

Lakhani, K.R., Wolf, R.G., 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects, in: Feller, J., Fitzgerald, B., Hissam, S.A., Lakhani, K.R. (Eds.), Perspectives on Free and Open Source Software. MIT Press, pp. 3 –22.

Lee, J.A., 2006. New perspectives on public goods production: Policy implications of open source software. Vanderbilt Journal of Entertainment and Technology Law 9.

Lerner, J., Pathak, P.A., Tirole, J., 2006. The dynamics of open-source contributors. American Economic Review 96, 114.

Lerner, J., Schankerman, M., 2010. The Comingled Code – Open Source and Economic Development. The MIT Press, Cambridge, Massachusetts.

Lerner, J., Tirole, J., 2002. Some simple economics of open source. Journal of Industrial Economics 50, 197–234.

Lerner, J., Tirole, J., 2005. The economics of technology sharing: Open source and beyond. Journal of Economic Perspectives 19, 99 – 120.

Linux Devices.com, 2007. Snapshot of the embedded linux market. `http://www.linuxfordevices.com/c/a/Linux-For-Devices-Articles/Snapshot-of-the-em`

Llanes, G., de Elejalde, R., 2009. Industry Equilibrium with Open Source and Proprietary Firms. Harvard Business School Working Papers 09-149. Harvard Business School.

Lohr, S., 2009. Software war pits oracle vs. google. The New York Times.

Maurer, S., 2012. The penguin and the cartel: Rethinking antitrust and innovation policy for the age of commercial open source. Utah Law Review 2012, 269–318.

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.

Miyagiwa, K., Ohno, Y., 2002. Uncertainty, spillovers, and cooperative R&D. International Journal of Industrial Organization 20, 855 – 876.

Motta, M., 1992. Cooperative R&D and vertical product differentiation. International Journal of Industrial Organization 10, 643 – 661.

Mustonen, M., 2003. Copyleft–the economics of linux and other open source software. Information Economics and Policy 15, 99 – 121.

O'Connor, A., Ong, K.W., Sander, T., Ferlo, M., 2005. Government Policies on Open Source. mimeo. University of Washington.

Osterloh, M., 2002. Open Source Software: New Rules for the Market Economy. Technical Report.

Osterloh, M., Rota, S., Wartburg, M.v., 2001. Open Source - New Rules in Software Development. Technical Report.

Pasche, M., Engelhardt, S.v., 2004. Volkswirtschaftliche Aspekte Der Open-Source-Softwareentwicklung. Jenaer Schriften zur Wirtschaftswissenschaft.

Raymond, E.S., 1998. The cathedral and the bazaar. First Monday 3. `http://firstmonday.org`.

Rolandsson, B., Bergquist, M., Ljungberg, J., 2011. Open source in the firm: Opening up professional practices of software development. Research Policy 40, 576 – 587.

Rosenkranz, S., 1995. Innovation and cooperation under vertical product differentiation. International Journal of Industrial Organization 13, 1 – 22.

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.

Schmidt, K.M., Schnitzer, M., 2003. Public subsidies for open source? Some economic policy issues of the software market. Harvard Journal of Law & Technology 16.

Schmidtke, R., 2006. Private Provision of a Complementary Public Good. CESifo Working Paper Series CESifo Working Paper No. 1756. CESifo Group Munich.

Scotchmer, S., 2004. Innovation and Incentives. The MIT Press.

Sen, R., 2007. A strategic analysis of competition between open source and proprietary software. Journal of Management Information Systems 24, 233 – 257.

Sharpe, M., 2009. Playing to win in the new software market – software 2.0: Winning for europe. report of an industry expert group on european software strategy.

Sorenson, O., Rivkin, J.W., Fleming, L., 2006. Complexity, networks and knowledge flow. Research Policy 35, 994 – 1017.

Välimäki, M., 2003. Dual licensing in open source software industry. Systemes d'Information et Management 8, 63 – 75.

Verani, S., 2006. Open Source Development in a Differentiated Duopoly. Economics Discussion / Working Papers 06-05. The University of Western Australia, Department of Economics.

Vogel, D., 2005. The Market for Virtue: The Potential and Limits of Corporate Social Responsibility. The Brookings Institution Press.

West, J., Gallagher, S., 2006. Challenges of open innovation: the paradox of firm investment in open-source software. R&D Management 36, 319 – 331.

West, J., O'Mahony, S., 2008. The role of participation architecture in growing sponsored open source communities. Industry and Innovation 15, 145 – 168.

Wong, K., 2004. Free/Open Source Software: Government Policy. Elsevier. Published by the International Open Source Network and UNDP Asia-Pacific Development Information Programme.

ZeroTurnaround, 2010. Java ee productivity report 2011. `http://www.zeroturnaround.com/wp-content/uploads/2010/11/Java_EE_Productivity_Rep`