Abstract

This paper is concerned with the economic trade-offs associated with open-sourcing, the business strategy of releasing free open-source versions of commercial software products. The effect of the release of open-source versions on the customers’ perception of products is an important determinant of open-sourcing outcomes. We model open-sourcing as a strategic option for firms that compete in the market for software products. Of particular importance in our model is the effect of open-sourcing on customer values and the possibility for better customization offered by the open-source products. We show that open-sourcing can arise as an equilibrium outcome in our simple two-stage game. If the enhancement of customer values from open-sourcing is moderate or high, firms may find it optimal to release open-source versions of their products.
1 Introduction

In the past decade, the development of open-source software (OSS) has received considerable attention from practitioners and academics alike. There is very little doubt that the process of developing free, useful and complex software initiated by the OSS movement has made consumers better off. Existing research on open-source has focused mainly on issues concerning the underlying motivation of programmers. Our interest here is of a different nature. We wish to investigate the observation – puzzling to many – that many firms in the current business environment choose to open-source some of their software products.¹

At first sight, open-sourcing can hardly be consistent with profit maximization. Clearly, the open-source and the commercial versions of a product are substitutes. It seems quite intuitive that by making a substitute product available free of charge, any software producer would lower its profit from the sale of the commercial product. Thus, we are quite uneasy about the tension that arises between this apparent reduction in profit and the observation that an increasing number of software firms choose to open-source their commercial products.

Initial interest about OSS from commercial entities has been materialized by adoption of an OSS product as a core component of a commercial product. An example is IBM’s adoption of the Apache web server as a core engine for their WebSphere product. Another example is Apple’s development of the Mac OS X operating system, which is based on the FreeBSD open-source operating system.

We have witnessed in the past few years an interesting and intriguing trend that constitutes, in a sense, the reverse side of OSS product adoption. An increasing number of firms release their products, free of charge. For instance, in October 2004, IBM released Cloudscape, a relational database product, to the Apache Software Foundation, an active member of the OSS community.²

Consistent with what seems to be the norm for this type of situations, IBM still offers customer support for the product they released to the OSS community. Other examples include the release by Sun Microsystems of OpenOffice Suite, a collection of office productivity programs that is derived from their commercial product StarOffice. Also, in August 2004, Computer Associates released their database product Ingres to the OSS community. Notably, in November of 2005, Computer Associates created a new company called Ingres Corp. to provide support and services for their

¹We refer to the release of a software product to the open-source community as open-sourcing.
²The OSS name of this IBM product is Derby.
database product.

The list of examples above is by no means meant to be exhaustive. The evidence points to an increasing number of firms that release “community editions” of their commercial software to consumers who can download the source code or executables and run them free of charge. We find this evidence rather intriguing. Why would firms that seem to enjoy a sizable stream of profit from the sale and service of a product choose to create or increase competition in their product market? How does open-sourcing affect the competitive environment faced by software firms? And, importantly, is there an economic mechanism through which open-sourcing can contribute to software firms enhancing their competitive position?

The following two quotes suggest some explanations. According to Prial (2004), IBM’s vice president of marketing and information management software,

“By open sourcing Cloudscape, IBM hopes to accelerate development of Java-based applications and drive more innovation around Linux and Java. [...] We think it will especially create new business opportunities [...]”

Bertrand Serlet, senior vice president of software at Apple, argues (see Serlet, 2004) that

“[With open-source code,] thousands of people look at the critical portions of source code and check those portions are right. It’s a major advantage to have open-source code.”

An increased pace of innovations – and thus, more added value for consumers – and improved security through increased exposure are, indeed, two of the major candidate explanations for the recent observed pattern of open-sourcing. But are these sufficient reasons to open-source a product? The answer is a qualified yes. Open-sourcing may be, in some cases, attributed to expected product innovation and improved security. Other reasons for open-sourcing have been mentioned in the literature, including the use of open-source products by firms who wish to gain an advantage over their competitors. Few of the explanations in the literature, however, discuss the impact of open-source products on the customers’ perception of the commercial and open-source products. We argue in what follows that the release of an open-source version affects the customers’ perceptions of a product, which in turn plays a very important role in a firm’s open-sourcing decision.
It is unlikely that the puzzle of open-sourcing can be explained away by employing a small set of economic arguments. It would be a significant departure from reality to expect that one can build a simple, one-size-fits-all economic model of open-sourcing. Instead of setting out to provide a comprehensive model of open-sourcing, we have a more modest – but also easier to achieve – goal. We intend to show by way of a simple model how open-sourcing can arise as an equilibrium business strategy. Even though our model is somewhat stylized, we are able to capture some of the principal economic trade-offs involved in the software developers’ decision to release open-source versions of their products.

We find that open-sourcing can be profitable in some situations. Open-sourcing can arise as a result of competition, despite the reduction in profit that is caused by “customer leakage” – i.e., the reduction in market share that may arise as a result of open-sourcing. We show that if the enhancement of customer values that results from open-sourcing is moderate, firms may find it optimal to release open-source versions of their products. When the value gains to the consumers from open-sourcing are high, we show that firms cannot take full advantage of these gains. The firms’ inability to funnel some of the customer value gains into higher profits is due to increased competition. Overall, our results indicate that, it is the consumers that are likely to benefit the most from open-sourcing.

The next section provides a review of the literature. Section 3 gives a brief outline of the market for open-source products. We develop our model in Section 4 and collect results in Section 5. Concluding remarks are in Section 6. Some of the more tedious calculations are relegated to the Appendix.

2 Related Literature

The recent developments associated with open source software seemed to have gained some notoriety. In turn, this has stimulated the interest of academics and practitioners. Early articles by Raymond (1999) about open source software have given rise to a wave of empirical and theoretical work. Schiff (2002) provides a comprehensive survey of the early literature on open source software. The current research can be classified into three broad categories (see von Krogh and von Hippel, 2006). Analysis of the motivations of open source contributions is by far the most popular.
research topic, perhaps because it has at its core the puzzling observation that cohorts of talented programmers choose to contribute to OSS projects with no apparent compensation. This stream of work includes empirical and theoretical work concerned with gaining an understanding of the motivations of OSS contributors (see Lerner and Tirole, 2002; Roberts et al., 2006; Bagozzi and Dholakia, 2006; Baldwin and Clark, 2006; von Hippel and von Krogh, 2003; Bitzer et al., 2007). Topics concerning the governance, organization and innovation process associated with OSS constitute the second main stream of OSS research. Unlike most commercial software projects, OSS project contributors are voluntary and located in various parts of the world. Thus, important issues highlighted by this stream of research include the challenges of managing OSS projects - mainly allocation of tasks and responsibilities of providing and reviewing bug-fixes, as well as the management of the process of innovation - issues dealing with scheduling product feature enhancements and product releases, associated with OSS projects (Koch and Schneider, 2002; Kuk, 2006; Mockus et al., 2002; MacCormack et al., 2006; Raymond, 1999; Stewart et al., 2006). The third stream of research is focused on the competition between open source and traditional, closed-source software. This stream of research includes analysis - both empirical and theoretical - of the public and free nature of OSS products and their impact on the marketplace for software products (Bonaccorsi and Rossi, 2003; Mustonen, 2005; West, 2003; Casadesus-Masanell and Ghemawat, 2006; Economides and Katsamakas, 2006; Bonaccorsi et al., 2006).

Our paper belongs to the third stream of OSS research. We seek to provide some economic explanations for the increased incidence of firms that compete by releasing open-source versions of their proprietary software products. Like our analysis, a few studies examine hybrid business models in the software industry that include proprietary and open source software (Krishnamurthy, 2005; Bonaccorsi et al., 2006). Krishnamurthy (2005) analyzes business models of firms that package, use or provide services for code produced primarily by the OSS community. Bonaccorsi et al. (2006) survey Italian firms that have combined proprietary and open source offerings under differing licensing schemes. They provide evidence indicating that firms are keen to adopt new hybrid models. Wichmann (2002) provides an early account of the motivations of large firms that participate in OSS activities. Some of the motivations of large firms suggested by Wichmann include the enhancement of a firm’s business prospects in a market of a complementary good (e.g., hardware or services), strategic considerations like the adoption of Java-based technologies and the need for product
standardization that could enable compatibility among various sub-systems, like the adoption of open document formats for word processing software. Lerner and Tirole (2002) view open-sourcing as the “razor" that is given for free (the code) to sell more “razor blades" (the hardware that is a complement of their code). They emphasize that open-sourcing is more likely to be a strategic tool of firms that are too small to compete or lagging behind in the commercial segment. Fosfuri et al. (2005) investigate the interest of profit-oriented firms in OSS products. Their empirical study highlights the importance of market position and technological capabilities in software and hardware on a commercial firm’s decision to introduce OSS products. They find that firms that are strong in proprietary software market and that are technologically competent in hardware are less likely to introduce OSS products while firms with high technological software capabilities and significant market power in hardware market are more likely to participate in OSS. Rossi and Bonaccorsi (2005) in their empirical study of Italian software producers highlight the extrinsic and intrinsic motivations of firms choosing to open their proprietary code. An important extrinsic motivation in their study is that firms that open code expect to obtain contributions and feedback in order to fix bugs and improve the software. Other explanations exist. As perhaps best articulated by IBM’s Jon Prial (2004), open-sourcing increases the rate of innovations, and thus it increases demand for a complementary commercial product of the same firm.

A few studies examine the competition between commercial software and OSS. Casadesus-Masanell and Ghemawat (2006) analyze the dynamic interaction in a vertically differentiated duopoly consisting of a profit-maximizing firm and a competitor that prices its output at marginal cost. Their model is inspired by the competition between Linux and Windows and emphasizes dynamic network externalities that arise as a result of demand-side learning. Similarly inspired by the competition between Windows and Linux, Economides and Katsamakas (2006) develop a framework for the two-sided pricing strategy of a software product developer whose product serves as a platform for complementary applications\(^3\). The studies of Casadesus-Masanell and Ghemawat, and Economides and Katsamakas assume the existence of an OSS product without specifically addressing the determinants of the firms’ decision to “release” their code.

\(^3\)Firms that control the development of a software platform make decisions on pricing to end-users as well as technical compatibility for the products of other developers that run on this platform. The authors call this two-sided platform pricing. For example, firms that develop game consoles set a price to the customer for the game console and often charge royalties to developers of games, following a two-sided pricing strategy.
Hawkins (2004) makes an important point by arguing that the release of code may be profitable because it entails a reduction in the cost of maintaining the code. The basic economic trade-off is between the increase in the buyers willingness to pay and the loss of market share that arises as a result of supporting the free substitute, as well as with the costs incurred by the firm in the process of supporting the free substitute software. Mustonen (2005) constructs a model in which a firm can choose to support a rival “copyleft free” software to gain compatibility. The firm’s decision not to support the rival software results in incompatibility between its commercial program and the freely available substitute. In this environment, compatibility is viewed as a way to increase the customers’ willingness to pay for the commercial product because of network effects. Mustonen’s model is similar to ours in that it considers consumers who are heterogeneous with respect to their valuations of the “copyleft” and “copyright” products, but his is a model in which only one of the firms acts strategically. A similar analysis is that of Sen (2007), who develops a model that explores the competition between proprietary software, an OSS product and a commercially-supported offering of the OSS product. Sen differentiates the three products in terms of their usability and shows that the presence of linkages between products affects the optimal level of usability that can be incorporated into a product. August et al. (2007) consider a model in which a firm chooses to adopt an OSS architecture. As a result, profits accrue to the firm that stream from services such as integration, support and consulting associated with the open source product. They analyze the impact of a competitive developer who enters the market to provide services for the open source product. They find that under some conditions, the developer may forgo profits from the product sales and go with open source strategy and rely purely on profits from the services market.

In our view, the explanations given in the literature for the existence of open source software – while providing valuable insight – go only some way toward identifying the reason why software firms choose to open-source their products. In the academic literature and in the media, two stories seem to coalesce as the most likely candidate explanations for open-sourcing. First, the release of open source products increases market size, so that firms benefit from the sale of complementary products or services. Second, the release of open source products reduces the cost of maintaining and debugging the code. The logic of both arguments relies on the fact that open-sourcing may be, from a dynamic perspective, profitable for a software firm. Both explanations rely on the intuition
that a favorable trade-off exists for the software firm between short run losses in revenue that stem from “customer leakage” (i.e., the reduction of revenue that arises as a result of making a substitute product available free of charge) and, in the long run, increased profitability that is due to higher revenues or to lower costs. We think that in this particular case intuition is misguided. If the main consequence of open-sourcing is an increase in the number of customers who use the product, market size could also be increased through free distribution of closed-source software. Free distribution could also result in better testing and bug-reporting. Furthermore, the reduction in the cost of maintaining or debugging the source code achieved as a result of “more eyeballs” scanning the released source code for bugs could be achieved through the release of the source code to a set of qualified firms or individuals, and not to the community at large.

It is unlikely that the economic reasons that drive a firm’s decision to release open source products could be easily enumerated. Most likely, a firm’s open-sourcing decision is affected by a multitude of factors. Some of these factors are identified in the literature. Our contribution is to bring to the fore an important, yet little explored aspect of open-sourcing: the impact of open source releases on customers. In the next section we turn to highlight some of the important characteristics of open source products and to explore the ways in which these characteristics affect the customers’ perception of open- and closed-source products.

3 OSS Market and Products

Hardly any online user forum devoted to a particular software product lacks complaints from users concerning the product features or, more often, the absence of desired features. Most software products – in their out-of-the-box state – fail to meet each minute requirements of their users. Closed-source products are not highly customizable. Conversely, OSS products may be customized to meet any user’s precise needs. OSS users may manipulate the source code, either to make minor modifications or to go as far as to significantly change the product’s functionality. While users of closed-source software products are generally restricted to making only minor changes to the product, they may modify their processes or practices in order to use the software more effectively. The time and effort spent incorporating desired functionality into OSS products or adapting to the requirements of closed-source products is reflected in costs incurred by the user. We believe that,
for most products, the cost of customizing an open source product is lower than the cost of adapting to the requirements – and customizing, to the extent possible – its commercial counterpart.\footnote{Customization costs clearly differ depending on the amount of customization needed. Our statement concerns the unit customization costs.}

Intimately related to open-sourcing is the issue of perception of OSS by the consumer. Some consumers may have a hard time assessing whether an open-source version has the same performance as the original, proprietary version (or a competing developer’s product).\footnote{In fact, comparing the original software to its open-sourced counterpart is problematic, since the code of the original, closed source version is never released. One can only compare the performance or feature set of the OSS product relative to the proprietary alternative.} For example, Star Office is an office productivity product developed by Sun Microsystems that is used primarily for word processing, spreadsheets and presentations. Its OSS equivalent, Open Office, offers the same core features. Consistent with what seems to be the industry norm, the closed-source Star Office provides more features than Open Office. These additional features comprise an integrated database product, licensed spell check and thesaurus, numerous clip arts, migration tools and a configuration manager for document filtering. The amount of customization afforded to a Star Office user is minimal. However, users of Open Office can change its freely available source code to modify or add functionality. Customers who value the higher base functionality of the proprietary product could be enticed to use its open-source counterpart because of its relatively easier customization.

SugarCRM is a provider of commercial open source customer relationship management software for companies with several deployment options to suit customers’ security, integration and configuration needs. They offer two distinct products: Sugar Enterprise and Sugar Community Edition. The community edition allowed users to view and change the source as long as they follow the Sugar Public Licence (currently GPLv3). Unlike the community edition which is free, the enterprise version can only be acquired for a fee. Users of the Sugar software could perceive other significant differences between the enterprise and the community editions. The Community Edition lacks the functionality required to create teams or to assign access levels to the teams (Farber, 2005). The inability to keep users from deleting each other’s contacts, schedules, leads, etc. makes the community version relatively unfit for commercial use. The missing functionality is added in the Enterprise edition. SugarCRM employs full-time developers and the new features incorporated into the commercial versions are generally missing from the free OSS version. Similar to the case
of Sun’s office products, the SugarCRM example indicates that the OSS version of the proprietary software product is “crimped” in that it has reduced functionality. The concept of crimping is not new. Deneckere and McAfee (1996) describe it in the context of Intel processors. Our software case is similar in that the commercial developer incurs a cost to provide the (OSS ready) lower functionality product. However, the analogy breaks down when we consider that savvy OSS users have the freedom to re-establish the “crimped” functionality by re-writing the relevant code. The free availability of the source code allows the user to make changes to the product at a cost that we believe is lower than the cost of changing the functionality of the proprietary version.

Our model formalizes these features of open-source products. We examine a market where the commercial version of a product provides more functionality (and thus, more value to the customers) than the OSS version. We analyze how the impact of open-sourcing on customers’ values affects the firms’ decision to open-source their products.

4 Model Setup

We consider the incentives for open-sourcing in a duopoly. The strategies of the firms in our model include a decision to open-source their output. Prior work analyzed the competition between an OSS alternative and a closed-source commercial alternative. Sen (2007) develops a model that explores the competition between proprietary software, an OSS product and a commercially-supported offering of the OSS product. This three-way competition framework in Sen’s paper is posited. In contrast, our model analyzes the competition between closed- and open-source products as an outcome of the firms’ strategic decisions to open source their products. Sen explores the effects of bringing commercial software into existence on prices, profits and the firms’ investment in software usability under various levels of network effects. His work exploits the differences in the levels of software usability across the three alternatives. The commercial open-source product has less usability than the proprietary product, but it is more usable than the free open-source version. In Sen’s model, the customers with the highest valuation for usability end up purchasing the proprietary software, while consumers with an intermediate valuation for usability purchase the commercial open-source product, and the rest use the free open-source software. Product differentiation in Sen’s model arises because of different product documentation and support services, not
because of different product characteristics. In contrast, our model assumes that differentiation is embodied in the product itself. In a recent paper, August et al. (2007) analyze the provision of services such as integration, support and consulting in an open-source setting. The model developed in their paper entails a choice between open- and closed-source architectures. In their model, upon choosing the open-source alternative, a firm enables a competitive developer to enter the market to provide services for the same product. Thus, in their model there is an inherent trade-off between the opening up of code and greater competition in the services market. While often software and service are hard to disentangle, we gain some modeling flexibility by focusing only on the product market. Both Sen (2007) and August et al. (2007) show that there may be benefits to open-sourcing when services are considered in conjunction with the software product. However, these papers do not focus on the firms’ incentives to open-source their products in the absence of competition in the services market. We further the analysis of Sen and August et al. by showing that open-sourcing may arise as an outcome of competition without explicitly considering the effect of the competitive forces in the services market.

The technical requirements of software vary greatly from consumer to consumer. Different consumers use the same software product in different ways to complete related, but potentially different tasks. Depending on the user’s ability, each task performed by the software requires some functionality. A single software product with a set of predefined features is unlikely to satisfy the requirements of all individual consumers. We use the concept of Task Technology Fit (TTF) introduced by Goodhue and Thompson (1995) to gauge the extent to which a given technology could assist an individual user in performing certain tasks. Depending on their required functionality of a software product, consumers will have varying levels of the TTF for that product. We model the variability of the TTF for a given software package in the population of consumers using a spatial model of product differentiation. We wish to keep our model simple, so we consider an environment with only two profit-maximizing firms. As it is commonplace in the literature on product differentiation, we assume that the two firms are located at the ends of a line segment of unit length and share a measure of consumers that, without loss of generality, we normalize to one. We also assume that the consumers are continuously (and uniformly) distributed over the unit length segment. We interpret the location of a consumer relative to a firm as that consumer’s ideal product functionality requirement. A consumer who is closer to a given firm incurs a smaller
disutility in terms of the TTF required to use the firm’s software than a consumer who is farther away. As such, we model consumers as heterogeneous in their fit for the functionality offered by the products of the two firms. We assume that the two firms are symmetric in all relevant attributes, except location. To operationalize the notion of task technology fit, we assume that consumers incur a specific unit fit cost \( \tau \) to use a given software. Given the symmetry of the firms, at equal prices, a consumer prefers the firm that is closer. In this sense, consumers located relatively close to a firm are “captive” and thus each firm enjoys some market power.

Figure 1: Market for a software product in duopoly

Figure 1 provides a depiction of our main setup. Let \( x \) denote the distance of a customer from Firm 1 on the unit line. The customer could purchase the output of either firm. We assume that consumers purchase at most one unit of output from either of the two firms. If the customer buys the closed-source product of Firm 1, the customer enjoys utility \( V - \tau x - P_1 \). If the customer buys the closed-source product of Firm 2, the customer enjoys utility level \( V - \tau(1 - x) - P_2 \).

Either firm has the option of open-sourcing its software product. We note that the closed-source product provides additional value through the use of proprietary features such as specialized tools, clip art, etc. Since these enhancements are available only to purchasers of the commercial version, the open-source version of the product lacks these sophisticated proprietary features.\(^6\) This reduction in value is quantified as \( \Delta_3 \). In addition, we recognize that users of both products (commercial and open-source) gain additional value from the availability of the source-code of the open-source product. It must be noted that the commerical version and the open-source version share the same code-base. Hence, the availability of the source code allows all users to inspect the source code and identify bug fixes or develop enhancements that are available to all users. This value is quantified as \( \Delta_1 \). Accordingly, we assume that the intrinsic value of the commercial product to a customer is \( V + \Delta_1 \). For simplicity we denote \( \Delta_2 = \Delta_3 - \Delta_1 \). Therefore, the intrinsic value of the OSS product is \( V - \Delta_2 \). We assume that \( \Delta_{1,2} > 0 \). Note that the term \( \Delta_1 + \Delta_2 \) represents the difference

\(^{6}\)Consistent with the StarOffice and SugarCRM examples discussed in Section 3.
in value associated with purchasing the open-source product over the commercial version. We view the open-sourced product as more customizable than the closed-source product. Since the OSS product is more easily customizable than the closed-source product, we assume that a customer’s fit cost for the open-source product is $\alpha \tau$ where $0 \leq \alpha \leq 1$. It follows that a consumer located at distance $x$ in product space from the first firm enjoys utility levels $UC_1 = (V + \Delta_1) - \tau x - P_1$, and $UC_2 = (V + \Delta_1) - \tau (1 - x) - P_2$ if the customer buys the closed-source product from Firm 1 and Firm 2 respectively. If the consumer chooses the open-source alternative of either firm, the consumer nets utility level $UO_1 = (V - \Delta_2) - \alpha \tau x$, or $UO_2 = (V - \Delta_2) - \alpha \tau (1 - x)$. Implicit in our definition of the open-source product is that the OSS products are offered free of charge by the two firms. For simplicity, we take as given in our model that all customers have the ability to costlessly install and use the open-source products.\footnote{Extensions are possible in which only a fraction of the customers has the ability to use the open-source product; we leave these for further research.}

4.1 Duopoly with closed-source products

The simplest case in our environment is that of two firms competing with closed-source products. Let $x_0$ denote the location of the customer who is indifferent between purchasing a closed-source product from either firm. All customers to the left of $x_0$ prefer to purchase the output of Firm 1, whereas the customers to the right of $x_0$ prefer to purchase the output of Firm 2. The profit functions for Firm 1 and Firm 2 respectively are: $\pi_{10} = x_0 P_{10}$ and $\pi_{20} = (1 - x_0) P_{20}$. Since the customer located at $x_0$ is indifferent between purchasing either product, we have $x_0 = \frac{P_{20} - P_{10} + \tau}{2\tau}$. In equilibrium, the two firms choose their prices $P_{10}$ and $P_{20}$ to maximize their profit, given that their opponent’s price is at the equilibrium level. Since the firms are symmetric, we are looking for a symmetric equilibrium that entails $P_{10} = P_{20}$. Fixing Firm 2’s price at its equilibrium level $P_{20}$, Firm 1’s profit as a function of its price $P$ is:

$$\pi_1(P) = P \frac{P_{20} - P + \tau}{2\tau}$$

maximizing with respect to $P$ and requiring that, by symmetry, the profit-maximizing price be equal to $P_{20}$ yields $P_{10} = P_{20} = \tau$. Intuitively, as customers incur a higher fit cost they become
more captive, and thus the firms enjoy more market power and could afford to increase their prices. The equilibrium profits of the two firms are $\pi_{10} = \pi_{20} = \frac{\tau}{2}$, consistent with the notion that more market power – indicated by higher customer fit costs – translates into higher profits for the two firms. Having established our benchmark, we turn next to an analysis of competition in which one of the firms also offers an open-source product.

4.2 Duopoly with only one firm offering an open-source product

Suppose Firm 1 decides to offer, free of charge, an open-source version of its commercial software product. The introduction of the open-source product has two main effects. First, some of Firm 1’s customers would find it more profitable to choose the free open-source product. This effect works so as to reduce the profit of Firm 1. The second effect entails making Firm 2 compete with the free open-source product of Firm 1. This essentially works so as to dampen the effect on Firm 1’s profits of changes in the price charged by Firm 2, and also as a way for Firm 1 to “steal” some of Firm 2’s customers. It is important to note that the customer who is indifferent between acquiring the product of either firm is contemplating a choice between the free open-source product of Firm 1 and the commercial closed-source product of Firm 2.

Analyzing competition in the presence of an open-source product is somewhat complicated because, depending on the model parameters, three configurations are possible involving varying measures of customers who acquire some of the three products. We depict the most general situation in Figure 2 below.

![Figure 2: Market for software products in duopoly where Firm 1 has open-sourced](image)

The customer located at $x_{11}$ is indifferent between the OSS product and the commercial product offered by Firm 1. The customer located at $y_{11}$ is indifferent between the OSS product (Firm 1’s) and the commercial product offered by Firm 2.
4.2.1 Interior Solution

We start with an analysis of the “interior solution” in which non-zero measures of customers choose each of the three products\(^8\). We use superscript \(I\) to identify the corresponding parameters. In this setting, all customers to the left of \(x_{11}^I\) purchase Firm 1’s closed-source commercial product, while all customers located to the right of \(y_{11}^I\) purchase Firm 2’s commercial product. The customers located between \(x_{11}^I\) and \(y_{11}^I\) find it optimal to use Firm 1’s OSS product. It follows that a fraction \(x_{11}^I\) of the customers purchase Firm 1’s commercial product and that a fraction \(1 - y_{11}^I\) of the customers purchase Firm 2’s commercial product. The remaining fraction \(y_{11}^I - x_{11}^I\) of customers chooses Firm 1’s OSS product. Given the demands for the three products, we can write down the two firms’ profits as \(\pi_{11}^I = x_{11}^I P_{11}^I\) and \(\pi_{21}^I = (1 - y_{11}^I) P_{21}^I\), where \(P_{i1}^I\) is the price charged by Firm \(i\).

To find \(x_{11}^I\) and \(y_{11}^I\) we need to set \(UC_1 = UO_1\) and \(UO_1 = UC_2\). Doing so, we find that:

\[
x_{11}^I = \frac{\Delta_1 + \Delta_2 - P_{11}^I}{\tau (1 - \alpha)}
\]

and

\[
y_{11}^I = \frac{P_{21}^I + \tau - \Delta_2}{\tau (1 + \alpha)}.
\]

Since in equilibrium the two firms choose prices to maximize profit, differentiating the two firms’ profits with respect to prices and solving for the two prices yields \(P_{11}^I = \frac{\Delta_1 + \Delta_2}{2}\) and \(P_{21}^I = \frac{\alpha \tau + \Delta_2}{2}\). In equilibrium, these prices give rise to values of \(x_{11}^I\) and \(y_{11}^I\) that can be expressed as:

\[
x_{11}^I = \frac{\Delta_1 + \Delta_2}{2 \tau (1 - \alpha)}
\]

and

\[
y_{11}^I = \frac{\tau (2 + \alpha) - \Delta_2}{2 \tau (1 + \alpha)}.
\]

\(^8\)In our setting an “interior” solution indicates that all \(x\) and \(y\) values are strictly between 0 and 1 and the most general ordering of the customer cutoff variables is maintained.
In equilibrium the profits of the two firms are:

$$\pi_{I1} = \frac{(\Delta_1 + \Delta_2)^2}{4\tau(1-\alpha)}$$

and

$$\pi_{I2} = \frac{(\alpha \tau + \Delta_2)^2}{4\tau(1+\alpha)}.$$  

We need to ensure that, according to our assumption, the parameters of our model are chosen so that $0 < x_{I1}^I < y_{I1}^I < 1$. It is readily verified that, given our choice of parameters, $x_{I1}^I > 0$ and $y_{I1}^I < 1$. To ensure sure that $x_{I1}^I < y_{I1}^I$, we require that:

$$\Delta_1 < \frac{(2 + \alpha)(1 - \alpha)\tau - 2\Delta_2}{1 + \alpha}. \quad (1)$$

When condition 1 is satisfied, there is a positive measure of customers who choose the free OSS version. Since some of these customers could have purchased the commercial product of Firm 1, we term the segment of consumers who choose the free OSS version “customer leakage.” It turns out that, depending on the choice of parameters, it is possible that customer leakage could be avoided altogether by the firm that releases an open-source version. We turn next to present an analysis of this situation.

4.2.2 No Customer Leakage ($y_{I1} \geq x_{I1}$)

If condition 1 is not satisfied, all customers prefer Firm 1’s commercial product to its free OSS version available. Intuitively, (1) is more likely to be violated if $\Delta_1$ or $\Delta_2$ – or both $\Delta_1$ and $\Delta_2$ – are relatively high, implying that the inherent value of the additional features offered in the commercial version is sufficiently higher than in the free OSS version. Note that the firms’ commercial offerings compete head-to-head in this case. To find conditions under which there is no customer leakage, we assume that while no customer finds it optimal to use the OSS version, Firm 1’s commercial product still benefits from the release of the open-source product (perhaps through a better management of code errors). Let superscript $II$ denote this region. The profits of the two firms are: $\pi_{II1}^I = x_{II1}^I P_{II1}$ and $\pi_{II2}^I = (1 - x_{II1}^I)P_{II2}^I$. Since the two commercial versions compete head-to-head, we find $x_{II1}^I$ by
solving for $x$ in $UC_1 = UC_2$ (note also that we need to check that $x_{11}^{II} \in (0, 1)$, so that Firm 2 still serves a fraction of the market). Straightforward calculations yield equilibrium prices chosen by the two firms that can be expressed as $P_{11}^{II} = \tau + \frac{\Delta_1}{3}$ and $P_{21}^{II} = \tau - \frac{\Delta_1}{3}$. Thus, in equilibrium

$$x_{11}^{II} = \frac{3\tau + \Delta_1}{6\tau}.$$

Note that Firm 1 enjoys a higher market share than in the benchmark case of section 4.1 because of the higher value that customers have for its product. The equilibrium profits of the two firms can be written as:

$$\pi_{11}^{II} = \frac{(3\tau + \Delta_1)^2}{18\tau} \quad (2)$$

and

$$\pi_{21}^{II} = \frac{(3\tau - \Delta_1)^2}{18\tau}. \quad (3)$$

If the value enhancement provided by the open-source product is large enough, Firm 2 may be driven out of the market entirely. We present an analysis of this case below.

### 4.2.3 Firm 2 is driven out of the market ($x_{11} \geq 1$)

We use superscript III to indicate the parameter region where $x_{11} \geq 1$. It can be easily checked that when the following condition holds, Firm 2 can no longer compete and Firm 1 becomes a monopoly:

$$\Delta_1 \geq 3\tau \quad (4)$$

Note that, as in the previous case, customers prefer the commercial version of Firm 1’s product to its open-source alternative. Thus, it turns out that to maximize its profit Firm 1 chooses price $P_{11}^{III} = \Delta_1 - \tau$ and has profit $\pi_{11}^{III} = \Delta_1 - \tau$.

Having exhausted the set of possible outcomes when one of the firms opens up its source code, we turn to an analysis of competition when both firms offer an open-source product.
4.3 Duopoly with open-source products

When both firms decide to open their products, there are four products in the market. The most general market situation is depicted in Figure 3 below in which non-zero measures of customers choose to purchase one of the four products.

![Figure 3: Market for software products in duopoly where both firms have open-sourced](image)

In Figure 3, the customer at $x_{12}$ is indifferent between the OSS product and the commercial product offered by Firm 1, while the customer located at distance $y_{12}$ from Firm 1 is indifferent between the two OSS products. In addition, the customer located at $x_{22}$ is indifferent between Firm 2’s OSS product and the commercial product of Firm 2.

4.3.1 Interior Solution

As above, we start with an analysis of the situation in which non-zero measures of customers choose each of the four products. Let superscript $I$ denote the corresponding parameter region. In this setting, all customers who are located on the left of $x_{12}^I$ purchase Firm 1’s closed-source commercial product and all customers located on the right of $x_{22}^I$ purchase Firm 2’s commercial product. The customers who are located between $x_{22}^I$ and $x_{12}^I$ use either Firm 1’s or Firm 2’s free OSS product. Note that these customers do not contribute to the profits of either firm, so their choice of Firm 1’s or Firm 2’s OSS product has no effect on the two firms’ profits. In this situation, Firm 1 sells its commercial product to a fraction $x_{12}^I$ of the customers, while a fraction $1 - x_{22}^I$ of the customers chooses Firm 2’s commercial product. A fraction of customers equal to $x_{22}^I - x_{12}^I$ chooses one of the two OSS products. Once again we refer to the segment of customers of measure $x_{22}^I - x_{12}^I$ as ‘leakage.’ Given these demands, the profit functions of Firm 1 and Firm 2 are $\pi_{12}^I = x_{12}^I P_{12}^I$ and $\pi_{22}^I = (1 - x_{22}^I) P_{22}^I$. Note that $x_{12}^I$ is determined by solving for $x$ when $UC_1 = UO_1$. Similarly, $y_{12}^I$ can be found by setting $UO_1 = UO_2$ and $x_{22}^I$ is found by solving for $x$ in $UO_2 = UC_2$. Straightforward calculations yield $x_{12}^I = (\Delta_1 + \Delta_2 - P_{12}^I)/(\tau(1 - \alpha))$, $y_{12}^I = 1/2$ and $x_{22}^I = 1 - (\Delta_1 + \Delta_2 - P_{22}^I)/(\tau(1 - \alpha))$. 
Differentiating the profit functions of the two firms with respect to their prices, and solving for the prices that jointly maximize the two firms’ profits yields:

\[ P_{12}^I = P_{22}^I = \frac{\Delta_1 + \Delta_2}{2}. \] (5)

The equilibrium values of \( x_{12}^I, y_{12}^I \) and \( x_{22}^I \) are \( x_{12}^I = \frac{\Delta_1 + \Delta_2}{2\tau(1 - \alpha)} \), \( y_{12}^I = 1/2 \) and \( x_{22}^I = 1 - \frac{\Delta_1 + \Delta_2}{2\tau(1 - \alpha)} \).

Hence, the equilibrium profits of the two firms can be written as:

\[ \pi_{12}^I = \pi_{22}^I = \frac{(\Delta_1 + \Delta_2)^2}{4\tau(1 - \alpha)}. \] (6)

Note that since the assumed solution entails non-zero measures of customers that use any of the four products, we need \( x_{12}^I < y_{12}^I < x_{22}^I \). Since the firms are symmetric, this translates into a single condition involving the two \( \Delta \)'s. It can be checked that if the following condition is to be satisfied in order for the parameters to yield such a solution:

\[ \Delta_1 < (1 - \alpha)\tau - \Delta_2. \] (7)

When this condition is violated, in equilibrium both \( x_{12} \geq y_{12} \) and \( y_{12} \geq x_{22} \). We derive the firms’ optimal pricing solution in this case next.

4.3.2 No Leakage (\( x_{22} = x_{12} \))

In this case, the two firms compete head-to-head with their closed-source commercial products. As above, we maintain the assumption that the release of the open-source product increases the value that consumers derive from using the commercial version of a product, even though no consumer could gain utility from using an open-source product. It turns out that, with or without this assumption, the equilibrium has the same properties as the equilibrium that we analyzed in the benchmark case above (so that \( x_{22}^{II} \) is equal to 1/2 and the profits of the two firms are equal, \( \pi_{12}^{II} = \pi_{22}^{II} = \tau/2 \)).

Having established the outcome of competition in all possible situations in our model, we turn next to an analysis of the incentives that firms may have to open-source their products.
5 Results

5.1 Unilateral Open-Sourcing

We analyze first the effect of open-sourcing on prices when only one of the firms chooses to open-source its code. We find that the open-sourcing firm will charge a lower price than its opponent only if there is customer leakage. The optimal solution entails customer leakage if the additional value that is due to open-sourcing (\(\Delta_1\)) is relatively small (i.e., if (1) holds). Recall that if this is the case, the open-sourcing firm cannibalizes some of its sales of the commercial product by issuing the free open-source product. The existence of a free open-source product implies that the open-sourcing firm will need to lower its price so as to reduce the extent of customer leakage. However, somewhat less intuitive is that when there is no leakage, as a result of open-sourcing, a firm will be able to increase the price it charges for its commercial product. The following proposition summarizes this result.

**Proposition 1.** When there is leakage the price charged by the firm that unilaterally opens its source code is lower than the price it would have charged had the firm not opened its source code. With no leakage, the price charged by the open-sourcing firm for its commercial product is higher.

*Proof.* Suppose Firm 1 unilaterally releases an open-source version of its product. If there is no leakage, in equilibrium the open-sourcing firm sells its commercial product at 

\[ P_{11} = \frac{\Delta_1 + \Delta_2}{2} \]  

(see (5)). We show next that this equilibrium price is less than the price in the benchmark case \((\tau)\) if the following holds:

\[ \Delta_1 < 2\tau - \Delta_2. \]  

(8)

Note that condition (8) implies that there is a restriction on the firm’s price, \(P_{11}^I < \tau\). We can easily show that (8) is satisfied when the solution is interior, as in section 4.2.1 above. To see this, observe that the right hand side of (8) is greater than the right hand side of (1) for all values of \(\alpha \in (0,1)\). Thus, in an equilibrium with customer leakage, open-sourcing forces Firm 1 to charge a lower price than in the benchmark case. With no leakage, depending on the values of parameters, either Firm 2 is active on the market (in which case the price charged by Firm 1 is \(\tau + \Delta_1/3\)), or Firm 2 is driven out of the market by the introduction of the open-source product (in which case
the price charged by Firm 1 is $\Delta_1 - \tau$). Evidently, in both cases, Firm 1’s price is greater than the benchmark price $\tau$.

Since in any interior solution the open-sourcing firm lowers its price relative to the benchmark, in order for open-sourcing to be profitable the open-sourcing firm’s market share has to increase to compensate the revenue loss on the customers it served in the benchmark equilibrium. This market share increase can compensate the revenue loss only for open-source products that bring about a relatively high incremental value gain $\Delta_1$. For small values of this incremental gain, open-sourcing is not a viable option. In the following Proposition we derive the minimum $\Delta_1$ that guarantees that unilateral open-sourcing is profitable.

**Proposition 2.** A firm will increase its profit by unilaterally open-sourcing if

$$\Delta_1 > \max\{0, \min\{\tau \sqrt{2(1 - \alpha)} - \Delta_2, \frac{(2 + \alpha)(1 - \alpha)\tau - 2\Delta_2}{1 + \alpha}\}\}. \quad (9)$$

**Proof.** Suppose that Firm 1 unilaterally releases an open-source version of its product. We first show when $\pi_{11} > \pi_{10}$ for the cases discussed in sections 4.2.1, 4.2.2 and 4.2.3. We start with the two corner solutions. When (1) does not hold, Firm 2’s market share can either be positive (as in section 4.2.2 above), or zero (as in section 4.2.3). When Firm 2 is active, $\pi_{11} > \pi_{10}$ is satisfied whenever $\Delta_1 > 0$. When Firm 2 is driven out of the market by the introduction of the open-source product, the first firm’s profit increases relative to the benchmark profit whenever $\Delta_1 > (3\tau/2)$. This condition holds true whenever Firm 2 exits the market (i.e., whenever $\Delta_1 > 3\tau$). Therefore, when $\Delta_1 > (2 + \alpha)(1 - \alpha)\tau - 2\Delta_2)/(1 + \alpha)$, a firm will have an incentive to open its source code unilaterally.

Suppose now that condition (1) holds, i.e., that $\Delta_1 < (2 + \alpha)(1 - \alpha)\tau - 2\Delta_2)/(1 + \alpha)$. If so, it is straightforward to show that $\pi_{11} > \pi_{10}$ whenever $\Delta_1 > \tau \sqrt{2(1 - \alpha)} - \Delta_2$ (see section 4.3.1 above). Combining the results yields the condition in (9).

The next question we would like to answer concerns the effect of open-sourcing on the profit of the firm that does not open its source code. We show next that open-sourcing unambiguously makes the opponent firm worse off.
Proposition 3. By unilaterally releasing an open-source version of its product, a firm makes its opponent worse off.

Proof. Suppose, as before, that Firm 1 unilaterally releases an open-source version of its products and that the solution is interior. We need to show that $\pi_{21} < \pi_{20}$. Suppose to the contrary that $\pi_{21} \geq \pi_{20}$. If so, we need $(\sqrt{2(1+\alpha)} - \alpha)\tau - \Delta_2 \leq 0$. Since, for $\alpha \in (0,1)$, $\sqrt{2(1+\alpha)} - \alpha > 1$, for $\pi_{21} \geq \pi_{20}$ to be true we require $\tau \leq \Delta_2$. Thus,

$$\frac{(2+\alpha)(1-\alpha)\tau - 2\Delta_2}{1+\alpha} \leq \tau(1-\alpha) - \Delta_2 \leq 0,$$

a contradiction, since (10) implies that the solution cannot be interior, as we assumed, since (1) is violated. Thus, Firm 2’s profit is lower in this situation. Evidently, Firm 2 is also worse off when it is driven out of the market. When there is no customer leakage and Firm 2 has positive market share, Firm 2 competes head-to-head with a higher-value product, Firm 2’s profit is reduced. This can be easily seen by inspection (compare (3) with the benchmark profit of $\tau/2$).

It is interesting to note that unilateral open-sourcing leads to higher surplus for all consumers, since both firms lower their prices.

We have shown that when one of the duopolists releases an open-source version of its product, its profit may increase. Furthermore, the release of the open-source version entails a reduction of the competing firm’s profit. The question then naturally arises, what is the outcome of competition when both firms release an open-source version of their products? We provide an answer in the next section.

5.2 Both Firms Open-Source

Suppose that the parameters of the model are chosen so that all four products in the market (two commercial products and two free open-source products) have positive market shares. As discussed above, this “interior” solution requires that $0 < x_{12}^I < y_{12}^I < x_{22}^I < 1$. Using the results presented in section 4.3.1 above, it can be verified that $x_{12}^I > 0$, $y_{12}^I < 1$ and $x_{22}^I < 1$ for all feasible parameter values. To ensure that all four products to have positive market shares, we also require that the parameters of the model satisfy the condition imposed by (7).

22
We provide next an analysis of the effect of open-sourcing on the prices charged by the two firms in equilibrium.

**Proposition 4.** The prices charged by the two firms in equilibrium are (weakly) lower when they both open their source code than without open-sourcing.

*Proof.* Suppose that the parameters of the model are chosen so that we are in the situation discussed in section 4.3.1. The equilibrium prices in that case are \((\Delta_1 + \Delta_2)/2\); it can be shown that prices are greater than \(\tau\) (the equilibrium price with no open-sourcing) if \(\Delta_1 > 2\tau - \Delta_2\). In turn, \(2\tau - \Delta_2 > (1 - \alpha)\tau - \Delta_2\) for all values of \(\alpha \in (0,1)\), so as long as (7) is satisfied, the desired result obtains. When (7) fails, as discussed in section 4.3.2, the equilibrium price is equal to the equilibrium price with no open-sourcing.

It is noteworthy that firms cannot increase their prices when they both open-source their products. We assumed that the commercial versions of the two products are more valuable to the customers when an open-source version is released. Despite the higher value that customers place on the products of the two firms, the additional value accrues to the customers alone, since in equilibrium the two firms do not profit from the release of the open-source version of their products. Any potential profit gains are lost due to competition. Firms not only compete with each other, but also compete with their open-source versions. The joint release of a free open-source version leads to lower prices for customers. However, since none of the firms is able to increase its market share, the firms’ profits cannot increase as a result of open-sourcing. The best the firms can do is to compete head-to-head when there is no customer leakage – in this situation the release of the open-source versions of their products has no effect on prices, market shares, and consequently, profits. We summarize this result in the following proposition.

**Proposition 5.** The two firms’ profits cannot increase as a result of both firms releasing open-source versions of their products.

*Proof.* When all four products have positive market shares, the profit for each firm is \(\frac{(\Delta_1 + \Delta_2)^2}{4\tau(1 - \alpha)}\) (see the calculations in section 4.3.1 above). Suppose that (7) holds. It is straightforward to check that these profits are smaller than \(\tau/2\) if \(\Delta_1 > \sqrt{2(1 - \alpha)}\tau - \Delta_2 := \hat{\Delta}_{15}\). In turn, \(\Delta_1 >
\[ \sqrt{2(1 - \alpha)\tau - \Delta_2} \] holds – for all values of \(\alpha \in (0, 1)\) – whenever (7) is satisfied, so profits are indeed reduced as a result of the two firms releasing open-source versions of their products.

When (7) fails there is no customer leakage and the equilibrium price and profits are unchanged from the benchmark case (see section 4.3.2 above).

Our results so far indicate that under some circumstances it is profitable for a firm to unilaterally release an open-source version of its product. Our results also suggest when both firms release open-source versions, their profits are reduced. Our model is predicated on the assumption that the two firms are symmetric. While it is possible that the observed pattern of releasing open-source versions of commercial products is driven in part by asymmetries between firms related to costs, consumer perception of the products, or the timing of the open-source releases, we wish to investigate next the outcome of competition when the two symmetric firms choose – simultaneously and independently – whether or not to release open-source versions of their products. We turn next to an analysis of the dynamic game induced by the firms’ open-sourcing and pricing decisions.

### 5.3 Open-Sourcing Equilibrium

We first describe the sequential-move game between our two firms. The game proceeds as follows: In the first stage, the firms independently and simultaneously choose whether or not to release open-source versions. In the second stage, upon observing their opponent’s open-sourcing decision, the firms, independently and simultaneously, choose their prices to maximize profit. Our equilibrium concept is subgame perfection (see Selten, 1975). A strategy profile for each of the two players is a subgame perfect equilibrium if it is an equilibrium in any of the subgames of the original game. We find the subgame perfect equilibria of our game using backward induction. We start with the second stage of the game. Depending on the firms’ actions in the first stage, there are four possible open-sourcing configurations. Only three of which are distinct, due to symmetry. The optimal pricing decisions and payoffs in each of these second stage configurations are discussed in Section 4. The three main cases of Section 4 provide the necessary payoff values for the first stage problem. Thus, we can evaluate the first-stage equilibrium outcomes using the payoffs we deduced in Section 4. We denote by \(\pi_{\text{neither}}\) the profit of each of the two firms when no firm releases an open-source version. The profit of each of the two firms when both firms release open-source versions are
denoted by $\pi_{\text{both}}$. In the asymmetric case when one of the firms releases an open-source version, we denote by $\pi_{\text{self}}$ the profit of the firm that released the open-source version and by $\pi_{\text{rival}}$ the profit of its opponent. Table 1 summarizes the payoffs that correspond to the first-stage actions of the two firms\(^9\).

<table>
<thead>
<tr>
<th>Firm 1</th>
<th>Firm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed</td>
<td>$\pi_{\text{neither}}$, $\pi_{\text{neither}}$</td>
</tr>
<tr>
<td>Open</td>
<td>$\pi_{\text{self}}$, $\pi_{\text{rival}}$</td>
</tr>
</tbody>
</table>

Table 1: Payoff Matrix Structure

The equilibrium outcome can be found by inspecting the firms’ payoffs. The outcome of competition depends on the choice of the model’s parameters since we have multiple solutions derived in sections 4.2 and 4.3. Thus, a different payoff structure may exist for different regions of the parameter space. We find that there are four different symmetric payoff matrices to be considered. We relegate the derivation of these payoff matrices to the Appendix. Table A-1 in the Appendix presents these payoffs.

Relevant in the computation of equilibria is the ranking of the firms’ payoffs in different competitive regimes. By choosing different values of the parameters of our model, the ranking of the profits that correspond to the first-stage actions of the two firms changes. Different equilibria obtain that correspond to the different ranking of the firms’ payoffs. We explore the parameter space in terms of the value of $\Delta_1$ – the incremental gain in the value of the commercial product brought about by the release of its open-source version – relative to the other parameters of the model. We find that the profits that result from the two firms’ first-stage open-sourcing decisions can be ranked differently depending on the value of $\Delta_1$ relative to certain cutoff values that are functions of the other parameters of the model.

We start by assuming that $\Delta_2$ and $\tau$ satisfy $\Delta_2 < \tau$. We turn to discuss next the ordering of the payoffs in the parameter regions. The ordering of the payoffs in the four regions is summarized in Table 2; a more detailed version of this table is provided as Table A-2 in the Appendix. We also relegate the definition of the cutoffs that define each region to the Appendix.

\(^9\)As we shall show, there are multiple payoff matrices to consider. This is due to the existence of multiple solutions.
Given these parameter regions and the ordering of the firms’ payoffs in each region, we can finalize our equilibrium analysis. Figure 4 summarizes the equilibria in each of the regions of the parameter space that correspond to Table 2. We note that multiple equilibria co-exist in some of the regions of the parameter space. In region A, since \( \pi_{\text{neither}} \geq \pi_{\text{self}} \) and \( \pi_{\text{rival}} \geq \pi_{\text{both}} \), the firms’ dominant first-stage action is not to release an open-source version. Regardless of its opponents’ action, each firm is better off with a closed-source product. Therefore, in this case (the benchmark discussed in section 4.1), in the unique equilibrium the two firms do not release open-source versions.

Figure 4: The equilibria with respect to \( \Delta_1 \) when \( \Delta_2 \leq \tau \).

The equilibria that correspond to parameters that fall in region B also contain outcomes in which the two firms do not release open-source versions. No firm would consider a release of an open-source version if its opponent were not to release an open-source version (since in this region \( \pi_{\text{neither}} \geq \pi_{\text{self}} \)). However, not releasing an open-source version is no longer the best action irrespective of the opponent’s open-sourcing decision. Given that the other firm has an open-source product, the best response would be to have an open-source product as well, \( \pi_{\text{rival}} \leq \pi_{\text{both}} \). Thus, opening the source code can also be part of the equilibrium. However, both firms are better off in the equilibrium that does not involve the opening of source code.

As above, in region C, there are two equilibria in which the firms either release or do not release open-source versions. Unlike the situation that arises when the model’s parameters fall within
region B, in region C the firms’ payoffs in both equilibria are the same (so the equilibrium that involves releasing an open-source version is no longer payoff dominated).

Inspection of the payoffs in Table 2 indicates that a firm could profitably open up their code when its opponent does not whenever the model’s parameters fall within region D. In this region \( \pi_{self} \geq \pi_{neither} \) and \( \pi_{both} \geq \pi_{rival} \). Thus, irrespective of the action of its opponent, a firm’s best first-stage action is to release its source code. Thus, the unique equilibrium has both firms releasing of an open-source version.

It can be easily seen by inspecting the values of the cutoffs that as the value of \( \Delta_2 \) increases relative to \( \tau \), the two regions A and B decrease in size. When \( \Delta_2 \geq \tau \) the regions I and II vanish. In that case, both firms open their source code in the unique equilibrium for all values of \( \Delta_1 \).

6 Conclusions

In this paper we analyzed the conditions under which firms find it optimal to release open-source versions of their products. Conventional wisdom suggests that open-sourcing increases the size of the market. As the argument goes, as a result of open-sourcing software products gain more exposure, which in turn allows firms to reap higher profits through either increased sales of complementary products (e.g., hardware) or through a reduced future cost of maintaining and managing the software code. This explanation is incomplete and somewhat fallacious, as it stands to reason that greater profit increases could be achieved through limited releases of the source code or through free distribution of the closed-source product. Recent research has considered the incentives for open-sourcing in connection with a complementary services market. While in today’s business environment the software and services markets are hard to disentangle, we gain some insight by focusing only on the software product market. The main driving force of our model is the impact of open-sourcing on the customer’s perception of the software products. Open-source versions tend to provide less functionality than their commercial versions. However, customers could find the open source version more valuable as a result of the better customization opportunities. In our model, the “crimped” product competes head-to-head with the products of the competing firm. As a result, the release of an open-source version better insulates a firm from the pricing strategy of its opponent. All things equal, this implies that the firm that releases the open-source version has
a competitive edge over its opponent. Clearly, the firm that unilaterally releases the open-source version increases its profit, provided that it can maintain its customer base. If there is customer leakage – i.e., when the release of the open-source version causes some of the releasing firm’s customers to migrate to the free, open version – the outcome is affected by the trade-off between higher prices and a smaller customer base. We have showed how these trade-offs affect the firms’ decision to release open-source versions of their products. We identified parameter regions in which the equilibrium specification for both firms in our model is to release open-source versions of their products. For other parameter regions, in equilibrium the two firms do not release open-source versions. Open-sourcing is likely to occur when the difference between the values of a customer who uses the proprietary version and a customer who uses the open-source version is high relative to the fit cost. Not all firms in today’s software business environment have open-sourcing in their strategic repertoire. However, it is comforting to note that in most examples in which open-sourcing arises in a competitive environment, there is a sizeable gap between the product valuations of the open- and closed-source products by customers.

The market for software products and services is under continuous evolution. Our model suggests that open- and closed-source software products are bound to co-exist. However, co-existence of the two types of products is more likely when the open-source product lacks significant features, or when the closed-source version becomes more valuable as a result of better code maintenance (like ridding the code of bugs). Also important is the ease with which customers could customize the open-source product. More facile customization of the open-source product implies that, all other things equal, an equilibrium is more likely to arise in which the firms release open-source versions of their software products.
References


APPENDIX

Derivation of the Payoff Matrices

We summarize, using Table A-1 and Figures A-1 and A-2, the regions of the parameter space that correspond to the possible payoff matrices. We find that there are at max four different symmetric payoff matrices to be considered. We report these payoff matrices in Table A-1.

We explore the parameter space in terms of the value of $\Delta_1$ – the incremental gain in the value of the commercial product brought about by the release of its open-source version – relative to the other parameters of the model. Recall that there were three cutoff values for $\Delta_1$ we derived in sections 4.2 and 4.3. Define the cutoff values given in inequalities (1), (4), and (7) that separate possible solution regions as:

$$\hat{\Delta}_{11} = (1 - \alpha)\tau - \Delta_2,$$
$$\hat{\Delta}_{12} = \frac{(2 + \alpha)(1 - \alpha)\tau - 2\Delta_2}{1 + \alpha}$$

and

$$\hat{\Delta}_{13} = 3\tau.$$

It is straightforward to show that $\hat{\Delta}_{13} > \hat{\Delta}_{11}$, $\hat{\Delta}_{13} > \hat{\Delta}_{12}$, and $\hat{\Delta}_{13} > 0$. We start by assuming that $\Delta_2$ and $\tau$ satisfy $\Delta_2 < \tau$. It can be also readily shown that, if $\Delta_2 < \tau$, we have

$$(1 - \alpha)\tau - \Delta_2 < \frac{(2 + \alpha)(1 - \alpha)\tau - 2\Delta_2}{1 + \alpha}.$$

As such, the cutoffs can be ordered according to:

$$\hat{\Delta}_{13} > \hat{\Delta}_{12} > \hat{\Delta}_{11} > 0.$$
Figure A-1: All feasible solutions with respect to $\Delta_1$ when $\Delta_2 \leq \tau$.

since in this situation it can be shown that

$$\frac{(2 + \alpha)(1 - \alpha)\tau - 2\Delta_2}{1 + \alpha} \leq (1 - \alpha)\tau - \Delta_2 \leq 0.$$ 

For $\Delta_2 \geq \tau$, the regions of the parameter space where the ranking of the firms’ profits is unchanged are depicted in Figure A-2.

Figure A-2: All feasible solutions with respect to $\Delta_1$ when $\Delta_2 \geq \tau$.

The figures also lists the corresponding profit figures for each region, derived in sections 4.2 and 4.3. Given these profit figures, we formulate four possible payoff matrices in Table A-1.

The payoff matrix at the top of Table A-1 contains the firms’ profits computed in sections 4.1, 4.2.1 and 4.3.1 that can obtain in region I. The second payoff matrix contains the firms’ profits that correspond to the situations considered in sections 4.3.2 that can obtain in region II. The third matrix is for region III and contains the firms’ profits corresponding to sections 4.1, 4.2.2, and 4.3.2. Finally, the payoff matrix at the bottom of the table uses the profits computed in sections 4.1, 4.2.3 and 4.3.2 and obtains when the parameters of the model are in region IV of the parameter space.
Table A-1: Payoff Matrices

<table>
<thead>
<tr>
<th>Region</th>
<th>Firm 2</th>
<th>Firm 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Closed</td>
<td>Open</td>
</tr>
<tr>
<td>I</td>
<td>Closed</td>
<td>$\tau/2, \tau/2$</td>
</tr>
<tr>
<td></td>
<td>Open</td>
<td>$(\Delta_1 + \Delta_2)^2 \over 4\tau(1-\alpha), (\Delta_1 + \Delta_2)^2 \over 4\tau(1+\alpha)$</td>
</tr>
<tr>
<td>II</td>
<td>Closed</td>
<td>$\tau/2, \tau/2$</td>
</tr>
<tr>
<td></td>
<td>Open</td>
<td>$(\Delta_1 + \Delta_2)^2 \over 4\tau(1-\alpha), (\Delta_1 + \Delta_2)^2 \over 4\tau(1+\alpha)$</td>
</tr>
<tr>
<td>III</td>
<td>Closed</td>
<td>$\tau/2, \tau/2$</td>
</tr>
<tr>
<td></td>
<td>Open</td>
<td>$(3\tau + \Delta_1)^2 \over 18\tau, (3\tau - \Delta_1)^2 \over 18\tau$</td>
</tr>
<tr>
<td>IV</td>
<td>Closed</td>
<td>$\tau/2, \tau/2$</td>
</tr>
<tr>
<td></td>
<td>Open</td>
<td>$\Delta_1 - \tau, 0$</td>
</tr>
</tbody>
</table>

Ranking of the Payoffs

We turn to discuss next the ordering of the payoffs in the parameter regions delimited by the cutoffs above. The ordering of the payoffs within regions III and IV stays the same. However, when the parameters of the model fall within regions I and II, the ordering of payoffs could change. Therefore, we may split these regions into two subregions a and b. There are two new cutoffs that we need to consider. These two cutoffs are shown on the horizontal axis in Figure A-3, below two of the feasibility cutoffs: $\hat{\Delta}_{14}$ and $\hat{\Delta}_{15}$. The first of these cutoffs is derived from the comparison of
\( \pi_{self} \) and \( \pi_{rival} \) in region I. We find that

\[
\pi_{self} \leq \pi_{rival}
\]

if

\[
\Delta_1 \leq \frac{(\alpha \sqrt{1 - \alpha \tau} - (\sqrt{1 + \alpha} - \sqrt{1 - \alpha}) \Delta_2)}{\sqrt{1 + \alpha}} := \hat{\Delta}_{14}.
\]

Thus, for relatively small values of \( \Delta_1 \), the opponent of the firm that releases an open-source version of its product can benefit from this release. Clearly, releasing an open-source version would not be part of the equilibrium in this case. Nevertheless, this cutoff has important implications for the existence of other equilibria, as we show below. Cutoff \( \hat{\Delta}_{14} \) is not necessarily contained in region I. If \( \hat{\Delta}_{14} \) is less than zero, then region Ia vanishes and \( \pi_{self} \) is greater than \( \pi_{rival} \). If \( \hat{\Delta}_{14} \) is greater than \( \hat{\Delta}_{11} \), then region Ib vanishes and \( \pi_{rival} \) is greater than \( \pi_{self} \).

The cutoff \( \hat{\Delta}_{15} \) is equal to \( \sqrt{2(1 + \alpha) \tau} - \Delta_2 \) and was used in the proof of Proposition 2. Recall that when \( \Delta_1 \) is greater than \( \sqrt{2(1 + \alpha) \tau} - \Delta_2 \), then a firm’s profit increases when it unilaterally releases an open-source version of its product. Since \( \hat{\Delta}_{15} \) falls between its adjacent cutoffs, it can be shown under our maintained assumption \( (\Delta_2 < \tau) \) that region II is split into two subregions. When \( \Delta_2 \geq \tau \), region II vanishes altogether as mentioned before.

The ordering of the payoffs in the four regions is summarized in Table A-2.

### Equilibrium Analysis

Given these six parameter regions and the ordering of the firms’ payoffs in each region, we can finalize our equilibrium analysis. For each of the regions of the parameter space delimited by the above cutoffs, Figure A-3 depicts all possible equilibria.

We note that multiple equilibria co-exist in some of the regions of the parameter space. In region Ia, since \( \pi_{neither} \geq \pi_{self} \) and \( \pi_{rival} \geq \pi_{both} \), each of the firms’ dominant action is not to release an open-source version. Regardless of its opponents’ action, each firm is better off with a closed-source product. Therefore, in this case (the benchmark discussed in section 4.1), in the unique equilibrium the two firms do not release open-source versions.
The equilibria that correspond to parameters within region Ib also contain outcomes in which the two firms do not release open-source versions, since neither firm would consider a release of an open-source version if its opponent were not to release an open-source version, since in this region $\pi_{\text{neither}} \geq \pi_{\text{rival}} \geq \pi_{\text{self}} = \pi_{\text{both}}$. However, not releasing an open-source version is no longer the dominant first-stage action. Given that the other firm has an open-source product, the best response would be to have an open-source product as well, $\pi_{\text{rival}} \leq \pi_{\text{both}}$. Thus, opening the source code can also be part of the equilibrium. However, both firms are better off in the equilibrium that does not involve the opening of source code.

As above, in region IIa, there are two equilibria in which the firms either release or do not
release open-source versions. Unlike the situation that arises when the model’s parameters fall within region Ib, in region IIa the firms’ payoffs in both equilibria are the same (so the equilibrium that involves releasing an open-source version is no longer payoff dominated).

Inspection of the payoffs in Table A-2 indicates that a firm could profitably open up their code when its opponent does not whenever the model’s parameters fall within regions IIb, III, and IV. In each of these three regions $\pi_{self} \geq \pi_{neither}$ and $\pi_{both} \geq \pi_{rival}$. Thus, irrespective of the action of its opponent, a firm’s best first-stage action is to release its source code. Thus, when the model’s parameters are in regions IIa, III and IV, the unique equilibrium has both firms releasing an open-source version.

We have maintained up to this point the assumption that $\Delta_2 < \tau$. We report the equilibria that arise when $\Delta_2 \geq \tau$ in Figure A-4.

\[ \Delta_1 \]
\[ \Delta_{13} \]

Figure A-4: The equilibria with respect to $\Delta_1$ when $\Delta_2 \geq \tau$. 

\[ 0 \]
\[ III \]
\[ IV \]