EUIPO
EUROPEAN UNION
INTELLECTUAL PROPERTY OFFICE

www.euipo.europa.eu

OPEN-SOURCE SOFTWARE IN THE EUROPEAN UNION

June 2020

OPEN SOURCE SOFTWARE IN THE EUROPEAN UNION

ISBN 978-92-9156-276-3 doi: 10.2814/866548 TB-03-20-304-EN-N

Project team:

Michał Kazimierczak, Economist, EUIPO

Altair Breckwoldt Jurado, trainee, EUIPO

Nathan Wajsman, Chief Economist, EUIPO

The authors would like to thank Catalina Martínez from the Spanish National Research Council (CSIC) and Alex Sander from the Free Software Foundation Europe for helpful comments on an earlier draft of this report.

EXECUTIVE SUMMARY

Regulation No 386/2012 entrusts the European Observatory on Infringements of Intellectual Property Rights (the Observatory) with tasks related to improving the understanding of the value of intellectual property rights (IPR) and fostering the exchange of information on new competitive business models which enlarge the legal offer of cultural and creative content. Within those remits, the Observatory prepares research reports that provide information on various contexts of IPR use and its importance to individual firms and the European economy. This report focuses on open-source software (OSS) as it is often regarded as an alternative way of organising innovation activity in the software industry.

The present report analyses the scope of use of OS licences by commercial firms in the European software sector and focuses on the way those firms use formal IP rights to sustain their business models involving software licensed under OSS terms.

OSS emerged as a reaction to the evolution of the software industry, the specificities of innovation in this sector and the uncertainties regarding IPR protection of software. The incremental nature of software innovation facilitates reuse of code. The importance of network economies promotes solutions increasing wide dissemination of software. The popularity of OSS has its roots in the cultural preference for software sharing among the early software developers' communities. Software also has some specific features which do not always fit well within the current system of IPR protection. This may result in shifting the business focus from software as such to products and services complementing software. It has consequences for the business models of software firms and the way they use IPR for sustaining those business models.

The main contribution of the present study is an analysis of the results of a survey conducted on a sample of 1 364 firms with seat in the EU and active in the software sector. This analysis shows that OSS has become an integral part of the business models of software firms. Most of these firms encourage their employees to develop or use software licensed under OSS licences during working hours. In recent years most firms taking part in the survey either maintained or increased their involvement in OSS development or use. OSS brings some tangible benefits, not limited to lower costs only. By embedding OSS software into their business models firms may get strategic benefits related to the wider dissemination of software and a larger knowledge pool they can tap into. The survey data shows that OSS does not limit but rather widens the business opportunities for software firms and most of them treat OSS as an opportunity rather than as a threat to their business model. The entry rate into industry is high and many newly created firms indicate the high importance of OSS for their decision to start a business. For more than one third of young OSS-developing or using firms, the existence of OSS was a crucial consideration for starting activity.

Firms allowing their employees to engage in OSS development during working hours are somewhat more sceptical as regards the current system of IPR protection than their counterparts less engaged in OSS. However, informal and formal IP protection methods are important for their ability to convert market opportunities associated with OSS into profitable business models. Survey responses show that the higher the share of revenue stemming from the sale of licences among OSS-developing firms, the more intense use of proprietary licences is made. Firms for which the share of revenue from the sale of licences is relatively lower are more likely to use non-proprietary licences for software distribution. Their incentive for contributing to OSS hinge, however, upon the protection of IP embedded in services and products

complementing OSS. Among those firms, the most popular strategy is based on the development of services complementing OSS or on embedding software in tangible products.

There are some differences in the use and assessment of the effectiveness of various measures of IP protection depending on the involvement in OSS. While non-OSS firms tend to rely relatively more on patents, trade secrets and designs, OSS-developing firms rely more on copyright, internet domain names and informal measures of IP protection. The use of trade marks is almost the same among different groups of firms. The difference between firms is not as evident in the assessment of the effectiveness of various IP protection measures. In fact, among OSS-developing firms the share of firms assessing copyright, trade marks and designs as effective measures of protection of their products and services is higher than among non-OSS firms. This suggests that firms engaged in OSS development are able to forego the proprietary licensing for software by relying on IP protection for other parts of their business models.

Analysis of the survey responses suggests that firms' engagement in OSS is driven by pragmatism related to its lower cost and the strategic advantages stemming from the decentralised process of OSS development, rather than by ideological motives. Development and use of software licensed under OS benefits firms by opening up new business opportunities based on services and products complementing OSS. It is also beneficial to OSS communities as the commercial interest of firms provides for more stable and long-term perspectives for OS software development.

OSS governance and the effectiveness of business models based on OSS rely heavily on the users' compliance with OSS licence terms and copyright provisions. This bears out the versatility of the current IPR system that enables the emergence of new innovation governance methods contributing to the wider dissemination of innovative products while preserving the ability of firms to benefit from their innovations.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The non-rival character of new ideas and knowledge provides clear economic advantages from their wide dissemination. Firms do not need to reinvent new ideas or technologies and they may become more productive by building upon ideas created by others (Romer, 1990). However, systems based on the open dissemination of knowledge cannot be sustained without providing mechanisms rewarding contribution to the common knowledge pool. The individual incentive is a crucial factor in the long-term success of any open technology system (Foray, 2006). The challenge that innovators face is not only the creation of value from innovation but also appropriation – capturing enough of this value for themselves to justify the investment in R&D. Insufficient appropriability will be detrimental not only to the innovator but to the society as a whole, resulting in a lower level of innovation activity. Private profits, at least at the level allowing recovery of the investment costs, are necessary to guarantee further investment in R&D and innovation (Pisano & Teece, 2007). Particularly in cases where the technology does not have strong natural barriers to imitation, IPR may be important for capturing the benefits from innovation.

Therefore, at first sight, the engagement of commercial companies in the development of OSS can be difficult to explain with the standard economic methods (Lerner & Tirole, 2002). The popularity of OSS, where access to the technologies and innovations created by others is open, the building blocks of previous innovations may be used and reconfigured for subsequent innovative projects, and commercial firms are able to design and employ innovative business models aligning those underpinnings with their profit motives, warrants further analysis.

For the purpose of this report, OSS has been defined as an application available together with its full source code, under a licence which allows anyone to use it and distribute for any purpose (including commercial use and distribution), both in the original and in a modified version (including derivative works) (1). However, OSS applications are not in the public domain. The OSS model depends on the traditional intellectual property regime, which ensures its viability through specific licence terms (Vasudeva, 2013). Whereas works in the public domain are completely free and may be used without any strings attached, some of the OSS licences are associated with important limitations as regards the use and distribution of the software and its derivatives.

Increasingly the past distinction between proprietary and OS software development modes has become less relevant, as many firms combine both methods of software development and distribution. As early as 2005 Jason Matusow, manager of the Shared Source Initiative at Microsoft, declared that ‘Few software companies are left that can properly call themselves either purely OSS (in the sense of OSS as a community-driven, not-for-profit exercise) or purely commercial’ (Matusow 2005). This creative mixing of proprietary/open modes of production results in faster innovation rates and cost savings but also gives

(1) The term ‘open-source software’ as used in this report overlaps with some alternatives used in the literature such as ‘free and open-source software’ (FOSS or F/OSS) or ‘free/libre open-source software’ (FLOSS). Chapter 5 explains the differences between the various subsectors of OSS based on the software licence terms and their compatibility with commercial activity.

OSS communities a more sustainable footing. It also calls for an analysis of the business models employed by software firms, the way of appropriating the value stemming from their innovation and the role of IPR in providing innovation incentives.

The present report focuses mainly on the analysis of the scope of use of OS licence models by commercial firms active in the European IT sector and the extent to which formal IP rights are employed to sustain business models that are partly based on open-source. To this end, the results of a survey conducted on a sample of 1 364 firms in 10 EU Member States, active in the software sector, are analysed. Software-related inventions are ubiquitous today and are implemented by firms active in many different industries. Some firms active in other sectors may therefore use and benefit from software developed under OSS licences. However, the survey has been limited to firms in the software industry, as firms from other sectors are not necessarily involved in software development, while all firms in the software sector are likely to be familiar with OSS.

Unique features of the software and software innovation process, evolution of software industry and the specific regime of IPR protection for software are important factors contributing to the rise in popularity of OSS solutions. Therefore, the quantitative analysis of the novel dataset generated by the survey is preceded by an extensive review focusing of those aspects of the software industry. Although the EU is the principal focus of the present analysis, this review also refers to other countries, primarily the United States, which was the country of origin of many important developments in the software sector.

The report will focus on answering the following questions regarding the role of open-source for the software sector in Europe.

- What is the level of engagement of for-profit software firms in the development of OSS?
- How has this engagement evolved in recent years?
- What are the most popular business models allowing software firms to profit from their engagement in OSS development?
- What is the role of IP protection for firms engaged in OSS development?
- How does the use of various IP protection tools differ depending on the intensity of OSS engagement?
- What are the most popular software licence types used by firms active in the software sector?

2. CHARACTERISTICS OF SOFTWARE AND THE SOFTWARE INNOVATION PROCESS

Software is a general term that is frequently used in relation to computer programs (Crowcroft, 2010). A computer program is an industrial product consisting of a set of instructions, which enable a computer to perform a series of tasks. Software is built upon lines of human readable code, which are referred to as the source code. The source code is the element that defines the functionality of the software, that is, specific tasks that can be performed.

A graphical user interface (GUIs) constitutes a software element that enables users to interact properly with it, shaping the user experience. A GUI can be considered as the look and feel of a computer program (Koukal, 2018) and constitutes the non-literal element of software. It has a 'look' aspect including features such as layout, colour and individual static components and the 'feel' aspect containing dynamic elements such as menus and buttons. A GUI allows for a more user-friendly interaction of the user with the software than interfaces based on a command line.

In many aspects, software resembles a machine (Samuelson et al., 1994); it interacts with hardware and other pieces of software to perform a task, and it is built using several components and assembly parts that, although not physical, must interact with each other. Failure to do so will make the software unusable.

Compatibility with other software is one of the most important strategic decisions at the development phase. Compatibility is a feature that allows users to run files produced in different software or to run software on a particular operating system (Slaughter, 2014). Compatibility is important because of the prominent role of network effects in the software sector. Katz & Shapiro (1985) defined network effects as the increase in utility derived by a user from the consumption of a particular good as the number of other consumers using the same type of good increases. Empirical research has confirmed that network effects stemming from a large installed base might be more important than product features for the market success of software (Brynjolfsson & Kemerer, 1996).

Compatibility is achieved by standardisation, that is, formal or informal agreements for the development and implementation of certain technical standards that allow for the interoperability of different software and hardware components. Standardisation encourages collaborative approaches, enabling development of novel functionalities and ensuring their compatibility with existing technical solutions (Husovec, 2018). Developers often rely on resources external to their organisations in order to deliver working solutions meeting customer expectations (Lippoldt & Stryszowski, 2009). In the software sector '[t]here are strong technical and economic reasons to reuse working solutions, and there is a strong cultural bias towards making this acceptable' (Crowcroft, 2010).

The software innovation process is cumulative and usually its life cycle is relatively short. Most of the revenues are realised in the short run after product release (Comino et al., 2015). Software production is also characterised by relatively high levels of R&D spending. R&D and development costs are high in the initial phase of software development, leading to the creation of the first working copy of the new software. However, the marginal costs of creation and distribution of additional copies are relatively low. In addition, software inventions have mostly a functional character and relate to the sequence and performance of computer instructions, whereas IP protection is well established and uncontested only for expressions of those inventions in the textual code. Those features of software create particular challenges for the

appropriability (ability of the innovator to retain sufficient added value it created with its novel products). As shown in the following section, the appropriability problem became especially apparent with the emergence of firms for which software was the main product.

3. EVOLUTION OF THE SOFTWARE INDUSTRY

Early computer programs were written for specific computers in a machine language, often by engineers building those computers (Fox, 2015). The release of the IBM System 360 family of computers in 1964 was an important milestone. As all the models in the System 360 family shared a similar standardised architecture, a software program could run on multiple machines (Graham & Mowery 2003; Slaughter, 2014). In those early days, software was developed by mainframe computer producers and their users: hobbyists, academics and researchers. Mainframe computer manufacturers sponsored groups of users sharing programs with one another, such as IBM's SHARE (Diaz, 2019). Sharing the source code was a widespread practice within those circles (William, 2002; Lerner & Tirole, 2002). Similar collective innovation practices and the active sharing of technical knowledge among direct rivals is not uncommon practice in the early phases of an industry's development, as illustrated by the examples of *Grand fabrique lyonnaise* (Foray, 2006) or by the development of the high-pressure steam engine in Cornwall in the early nineteenth century (Nuvolari, 2004).

With further evolution of the IT industry and the appearance of personal computers (PCs), separate software firms, independent from the computer producers, emerged. The computer industry became more 'vertically' organised with different firms specialising in different layers of computer systems. Evolution from the 'horizontal' to 'vertical' industry architecture was possible due to the modular character of PCs (Pisano & Teece, 2007). Standardisation facilitated benefits from economies of scale. Software became an important competitive asset, and starting in the late 1970s the industry saw explosive growth in the software firms that exploited many new niches emerging thanks to the growing user base. Due to rapid dissemination and the importance of network effects in the industry, some software products became de facto standards. The high number of users elevated the status of the providers of popular software to the position of dominant market players and raised the costs for customers who were considering switching to alternative solutions. The importance of such popular software for firms' revenue and the relative ease with which software could be copied illegally increased the importance of IPR protection for software products (Bonaccorsi et al., 2006; Graham & Mowery, 2003). Software firms started to actively assert their IP rights, providing software copies in binary code and limiting the possibilities for users to tinker with the software.

In recent years, the most important factor driving the evolution of the software industry has been the growth of networking technologies. It enabled new software delivery models such as click to download, software renting or software as a service (SaaS). It also had a huge impact on the demand for programming services driven by the strong increase in e-commerce and web-based business models affecting almost all sectors and economies. This also introduced important changes in the business models of the software firms. In the opinion of some authors, recent changes in the software industry may undermine the viability of stand-alone software products (Haff, 2018). Software is increasingly being paid for indirectly as an important part of a bundle, involving tangible products or by sophisticated means of monetisation based on users' attention and the data generated. Already in the early 2000s, Carr (2003) foresaw that with the advent of the internet many firms would fulfil their IT requirements by purchasing IT services online for a fee. The growth of the internet also provided an impetus for open-source popularity, as it enabled the work of many individual developers on the same software project to be decentralised.

These milestones in the evolution of the software industry were accompanied by a changing legal context and the adaptation of the intellectual property rights system to the new technologies. That is the subject of the following section.

4. INTELLECTUAL PROPERTY IN THE SOFTWARE INDUSTRY

Traditionally, intellectual property rights are regarded by economists as legal instruments providing incentives for innovation by allowing inventors to benefit from the temporary exclusivity in commercial exploitation of their innovative efforts (Scotchmer, 2004; Posner, 2005). Firms may use formal or informal instruments of protection to enhance their appropriability. However, the eligibility, scope and efficiency of various instruments of protection depend on many factors, such as the nature of invention in the sector or characteristics of the protected products and, therefore, differ between sectors. With the rise of firms focused on software development, a need appeared for the protection of intellectual property involved in software. Software represents a challenge for IP legislation worldwide as 'traditional' IP legislation was developed for the protection of physical goods or artistic works (Harison, 2008). The novel characteristics of software inventions made them difficult to fit into the existing system of IP protection. New *sui generis* methods tailored for the specific features of software were discussed in the 1970s and 1980s in Japan, France and in international forums such as WIPO, but were eventually rejected. In fact, the system of software protection developed based on the first decisions of IPR offices and courts in individual precedent cases (Kretschmer, 2003). Today, software protection is secured via a multi-layered system comprising traditional IPR, contract law and technical measures (Vasudeva, 2003).

4.1 ROLE OF COPYRIGHT PROTECTION

Council Directive 2009/24/EC of 23 April 2009 on the legal protection of computer programs defines the minimum level of software copyright protection in the Member States. Article 1 of the Directive explicitly states that *'Member States shall protect computer programs, by copyright, as literary works within the meaning of the Berne Convention for the Protection of Literary and Artistic Works'.* Some aspects of copyright protection, such as the protection of technological measures and rights-management information for software in the EU are also harmonised by the Information Society Directive (Directive 2001/29/EC). The duration of the copyright protection is set by Directive 2006/116/EC on the term of protection of copyright and certain related rights.

Copyright is also a standard method of software protection on the international level. The Berne Convention for the Protection of Literary and Artistic Property is the basis for the international protection of software and is widely accepted as such by the parties to that convention. This is reflected in the TRIPS Agreement[(2)], which in Article 10(1) explicitly stipulates that '*Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention'*. The WIPO Copyright Treaty[(3)], in Article 4, establishes that '*Computer programs are protected as literary works within the*

(2) The Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS) negotiated and administered under the auspices of the World Trade Organisation.

(3) World Intellectual Property Organization Copyright Treaty adopted in Geneva on 20 December 1996.

meaning of Article 2 of the Berne Convention. Such protection applies to computer programs, whatever may be the mode or form of their expression.'

Copyright protects not only software code but is also an appropriate legal instrument of protection of the GUI. In the judgment in case BSA, Ministerstvo kultury[(4)], the CJEU ruled that a GUI does not constitute a form of expression of a computer program within the meaning of the Software Directive and consequently cannot be protected specifically by copyright in computer programs by virtue of this directive. This does not preclude, however, the possibility to protect the GUI under the ordinary copyright law as per Directive 2001/29. Standard requirements regarding originality and a lack of protection for functional elements apply. In fact, although other forms of GUI protection, such as designs and trade marks, are available in some jurisdictions including some EU Member States, copyright is the most common type of protection for GUIs available in most countries (WIPO, 2016).

In accordance with the idea-expression dichotomy principle, copyright protection covers only expressions of ideas and not the underlying ideas themselves. This principle is designed to prevent the monopolisation of ideas and excessive rewards to authors. Accordingly, copyright protects the written form of computer programs (source code). The functionalities of computer programs, programming languages, data formats or architectural features of software are excluded from copyright protection[(5)]. Although protection provided by copyright may be effective for preventing large-scale illegal copying of software (Strowel & Utku, 2016), it does not provide sufficient protection against rewriting of innovative algorithms with different expressions (Martinez, 2018).

In contrast to traditional literary works, the expression of ideas is the least creative part of the software (Crowcroft, 2010). Due to the low expressive value of software, many scholars and judges describe copyright protection on software as rather 'thin' since computer programs are much more functional than works of literature (National Research Council, 1991; Shemtov, 2017). Additionally, a particular computer program may perform the same functions as another program, without literal copying of the underlying code and hence without infringing on the previous copyright[(6)].

The possibility to accommodate independent inventions is a feature of copyright protection, which differentiates it from patent protection, where to some extent protection of the functional elements of software code is possible as discussed below. Furthermore, although the general principle of the idea-expression dichotomy is widely accepted, the exact borderline between the two can be difficult to establish in concrete cases.

These particularities of software and the weakness of copyright protection related to the low expressive value of software have an important impact on the way software is provided to users. Until the emergence of the software as a service (SaaS) model, computer programs were usually released in binary code, which made it much more difficult for humans to analyse the ideas and functional concepts, which are not protected by copyright. Additionally, further terms limiting possible uses of computer programs are stipulated by contracts/licences.

---

(4) Judgment of 22 December 2010, C-406/10. World Programming Ltd.

(5) Judgment of 2 May 2012, *SAS Institute Inc*. *v World Programming, Ltd.*, C-406/10. World Programming Ltd., paragraph 46.

(6) Judgment of 2 May 2012, *SAS Institute Inc*. *v World Programming, Ltd.,* C-406/10. World Programming Ltd., paragraph 41.

4.2 PATENTS FOR SOFTWARE

The scope of patent protection may cover some embodiments of program functionalities. Due to the high turnover of software programmers, patents may be also a more secure way of innovation protection than trade secrets. At least theoretically, patents therefore provide much broader scope of protection of concepts related to software than do other IPR.

The interpretation of patent protection eligibility for software and its scope and breadth in Europe and the US has been constantly evolving during the past decades. To be patented, an invention has to meet two thresholds: eligibility and patentability. The eligibility threshold is higher in Europe than in the US, as Articles 52(2) and (3) of the European Patent Convention[(7)] explicitly exclude patents for programs for computers *as such*. This formulation leaves, however, some room for interpretation and does not preclude the eligibility of mixed inventions, consisting of both eligible and non-eligible subject matter (Chandler, 2015). This reasoning has been confirmed in the rulings of the EPO Board of Appeals. In the Vicom case[(8)], the Boards of Appeal decided that even though an invention may consist of methods for the digital processing of images, a claim directed towards a 'technical process which process is carried out under the control of a program (…) cannot be regarded as relating to a computer program as such within the meaning of Article 52(3) EPC'. After the Vicom ruling, software inventions were treated as patentable subject matter in cases where the claims comprised a technical contribution (Davies, 2003). This interpretation, often called *'any hardware approach'*, assumes that in cases where the claim envisages use of any piece of hardware, Article 52(2) does not apply. The *technical* character of the invention may be judged by assessing whether there is an effect on a technical process outside of the computer or inside the computer, such as increased speed or reliability (Chandler, 2015). However, a method concerning the use of technical means for processing purely non-technical information is not sufficient to meet the eligibility threshold[(9)].

Once the eligibility threshold is met, the examiner has to assess the patentability requirements. If the differences between the invention and the prior art are of non-technical character only, the patentability of such an invention may be discarded under Article 56 of the European Patent Convention due to an insufficient inventive step. So even though the Boards of Appeal interpretation of Article 52(2) for software-enabled inventions may be more liberal, legal hurdles related to the inventive step may still be difficult to overcome in the case of software patents. As summarised by Shemtov (2017) '*it is the inventiveness hurdle that effectively "filters out" non-technical subject matters as listed under Article 52, rather than Article 52 itself*'. The incremental character of much software innovation may therefore posit an important barrier to patentability in Europe. Nevertheless, Frietsch et al. (2015) estimate that between 2002 and 2010, over 35 % of total filings at the EPO could be categorised as computer-implemented inventions (CII)[(10)].

There is no direct mention of computer programs as not eligible subject matter in the US patent legislation. In spite of that, initially USPTO advocated against patent protection for CIIs invoking Section 101 of the Patent Act, which defines subject matter eligibility and contains an outright prohibition of claims comprising

(7) Convention on the Grant of European Patents of 5 October 1973.

(8) T-208/84 of 15/07/1986.

(9) T-931/95 Pension Benefit Systems.

(10) CII has been defined by authors as 'any invention implemented on a computer or similar apparatus, which is realized by one or more computer programs and which has at least one new feature implemented with the computer program(s).'

laws of nature, natural phenomena and abstract ideas (Diaz, 2019). Initial jurisprudence of the US Supreme Court, embodied for instance in the case *Gottschalk v Benson*[(11)], confirmed the stance of the USPTO. However, a 1981 Supreme Court decision *Diamond v Diehr*[(12)] opened the way for software patents. In that decision, the court posited that the mere presence of an algorithm does not automatically render a process or machine ineligible for patenting (National Research Council, 1991). Nevertheless, software patenting did not become widespread until the Federal Circuit Court's *re Alappat*[(13)] decision in 1995, which led the USPTO to issue the Examination Guidelines for Computer-Related Inventions a year later.

The scope of patentability of the computer-related inventions that are eligible for US patents has been clarified in recent years, as a result of the three Supreme Court decisions in cases *Bilski*[(14)], *Mayo*[(15)] and *Alice*[(16)]. These decisions are seen as largely limiting the scope of eligibility of software-related patents.

Recent jurisprudence of the EPO's Technical Board of Appeal and the US courts has resulted in a growing alignment as regards patentable subject matter of software patents between the EU and the US (Shemtov, 2017, Strowel & Utku, 2016). In both jurisdictions, a grant of patents for software *as such* may be problematic. However, the eligibility of CII may be assessed using different arguments. While examination at the EPO would concentrate on the determination of the technical aspects of the claims, the USPTO would concentrate on the question as to whether claims are related to *'abstract concepts'*.

The changing interpretation of the conditions for patentability of software-related inventions contributed to uncertainty of software patenting, which became a worldwide phenomenon (Li, 2019). This poses additional challenges for the applicants and legal practitioners (Strowel & Utku 2016). Immediately after the *Alice* decision, 830 software-related patents were withdrawn at the USPTO (Gray-Le Coz & Duan, 2014). In cases challenging patents brought to courts under Section 101 and explicitly citing the *Alice* decision, the invalidation rate exceeded 66 %. Among the cases examined by the Federal Circuit, only 3 out of 37 patents were upheld by the court, which results of invalidation rate of almost 92 % (Tran, 2016).

However, despite such uncertainty about the legal validity of certain granted software patents, firms still may have strategic reasons to apply for patent protection. Patents may be used as a defensive tool rather than as an incentive for innovation. As is the case in other sectors characterised by rapid and cumulative technological innovations, patents may become valuable 'bargaining chips' to achieve better outcomes in the potential licensing negotiations and can be used as a retaliation instrument in case of legal proceedings (Hall and Ziedonis, 2001; Graham et al., 2009). The results of the survey conducted by Frietsch et al. (2015) seem to confirm the strategic character of software patenting. Freedom to operate was the motive assessed as the most important one by the owners of CII patents filed at the EPO.

Due to some persistent differences in the interpretation of software patent eligibility, the European Commission made an attempt to harmonise the approach to the patentability of CIIs in Europe through a directive. Harmonisation is important because although the process of patent granting may be centralised at the EPO, patent enforcement rests with national courts and national courts practice may differ between

(11) *Gottschalk, Acting Commissioner of Patents v Benson et al.,* 409 U.S. 63 (1972).

(12) *Diamond v Diehr,* 450 U.S. 175 (1981).

(13) *Re Alappat,* 33 F.3d 1526 (Fed. Cir. 1994).

(14) *Bilski v Kappos,* 561 U.S. 593 (2010).

(15) *Mayo v Prometheus,* 566 U.S. 66 (2012).

(16) *Alice Corp. v CLS Bank International,* 573 U.S. 208 (2014).

EU Member States. However, the draft directive proved to be one of the most controversial pieces of legislation in recent EU history (Haunss, 2013). The legislative process was abandoned when the European Parliament voted against the adoption of the draft directive with the overwhelming majority of 648 to 14. This rejection was made possible by the mobilisation of many associations and civil society groups opposing the draft directive. One of the main arguments of those groups was that the incremental and combinatorial character of software innovation may lead to patent ownership fragmentation and emergence of *patent thickets* blocking future innovation and knowledge dissemination (Aigrain, 2010).

4.3 OTHER FORMS OF IPR PROTECTION FOR SOFTWARE

Given the fact that a revealed source code may be subject to a relatively easy reverse engineering process, firms that want to ensure the protection of their software rely heavily on trade secrets protection.

According to Article 30 of TRIPS, a trade secret must fulfil the following conditions: the information must be secret and must not be common knowledge for people related to that area, it must have some commercial value and the rightful owner must have taken reasonable steps to protect the secrecy of the information.

For the software industry, trade secrets can be an attractive form of protection for features such as content representation, algorithms and data structures. The potential attractiveness of this form of protection may increase with the rising popularity of SaaS and cloud computing, which enable access to software services without the necessity to reveal the code.

In the EU, trade secrets protection is regulated by Directive (EU) 2016/943 of the European Parliament and of the Council of 8 June 2016 on the protection of undisclosed know-how and business information (trade secrets) against their unlawful acquisition, use and disclosure, adopted by the European Council on 27 May 2016. The directive aims to guarantee a minimum standard level of protection, which should enhance competitiveness and improve cross-border collaboration in the development of intellectual assets. The definition of trade secret under the directive is modelled upon TRIPS. Among the important aspects of this legislation are the definition of unlawful and lawful acquisition as well as reverse engineering. Lawful acquisition takes place through independent discovery, reverse engineering using the information available to the public and other honest means. Unlawful acquisition requires that the information is secret at the time of acquisition and it is carried out without the consent of the trade secret holder or through any dishonest practice.

In the US trade secret protection is regulated by the Uniform Trade Secrets Act (UTSA) adopted in 1979. In 2016, President Barack Obama signed the Defend Trade Secrets Act (DTSA) with the aim of supplementing state laws. Both EU and US legislation deals with the concepts of trade secrets, misappropriation and the role and consequences of reverse engineering. US legislation, unlike the EU provisions, foresees the possibility of protecting software against reverse engineering by including confidentiality clauses in the respective licensing agreement.

EU law allows for design protection for some elements of GUIs. Graphical symbols and typographic typefaces are expressly listed in Article 3 of Council Regulation No 6/2002 of 12 December 2001 on Community designs as examples of products subject to design protection. The protection may be granted for GUIs that are novel, have an individual character and whose appearance cannot be solely dictated by their technical function. The universal principle is to allow for the protection of non-functional elements only.

The objective for such an exclusion is to avoid the monopolisation of certain technical functions through exclusive design rights[17].

Elements such as the overall layout, static components or animated features of GUIs can be eligible for trade mark protection. GUIs are eligible for trade mark protection both in the EU and the US as long as they fulfil the requirement of distinctiveness and non-functionality.

There are other modes of protection available to software firms, even if they are not directly available for software as such. Given the importance of the network externalities in the software industry, trade mark protection may prove to be especially important, as some firms may ride the coattails of the brands of market leaders. Registering a trade mark for a software name or logo is an efficient way of preventing competitors from applying a similar name to similar products.

4.4 ROLE OF LICENCES IN THE SOFTWARE INDUSTRY

Despite the fact that a piece of software may be protected by copyright, patents, designs or trade secrets, the appropriability regime for software innovation is not as strong as for other types of innovation. With access to the source code, it is relatively easy to reproduce the functional features of a piece of software while avoiding a literal copying of the code, and there is uncertainty as to what extent such functional features of program code may be protected by patents. Given this relative weakness of the protection provided by IP law, for many years licensing has been the dominant software transaction model, complementing public law provisions to protect the IP embedded in computer programs.

Given the incremental character of software innovation, licences are important for coordinating the development work of many independent entities on the same complex computer program. Due to the necessity to ensure the compatibility of software with hardware and with other computer programs, there is a need to ensure licensing efficiency. Fair, reasonable and non-discriminatory (FRAND) licensing terms for standard essential patents (SEP) and cross licensing patents within so-called patent pools are some of the mechanisms that can improve coordination among many entities. In practice, FRAND is a controversial concept mainly due to the multitude of SEPs, the fragmentation of SEP holders and implementers, and difficulties in determination of fair and reasonable royalties (Mennier&Thumm, 2015). There are also questions regarding the compatibility of FRAND licenses with certain types of OSS terms, especially strong copyleft licences (Husovec, 2018).

Licensing software has also been the preferred method of software products distribution. Under standard end-user licence agreements (EULAs), the buyer does not purchase the underlying intellectual property rights but rather a limited permission to use a copy of the software (Gomulkiewicz, 2018). By resorting to contract law and formulating appropriate terms in the licence agreement, software firms might purport to extend the protection conceded by public law, especially where the interpretation of the law is not sufficiently clear (Winston, 2006). Due to the widespread use of licences, some scholars treat the private law of licences in the software industry as an important legal regime in its own right (Phillips, 2009). The role of model licence terms in sustaining business models in the software industry has been emphasised by a former senior corporate attorney at Microsoft, who stated that '[f]or over two decades, we could say

(17) https://www.worldtrademarkreview.com/portfolio-management/graphical-user-interfaces.

that the licence is the product--software provides the functionality but the licence provides what can be done with the software' (Gomulkiewicz, 2018).

In a typical case, EULAs are not negotiated individually (they are classified as adherence contracts) and are accepted through user behaviour such as opening a box containing a copy of the software or clicking an *accept* button before downloading a program. Under EULAs, copies of the software are usually distributed in the form of binary code, without access to the source code. EULA licences precisely define users' rights and obligations. Breach of the licence terms might lead to the termination of the user's permission to use the software.

The enforceability of some of the licence terms that extend the protection beyond the limits set by public law is, however, doubtful from the legal point of view. Prohibition of computer program decompilation may be seen as equivalent to granting the rights holder the exclusive rights over elements that are protectable and those that are not protectable under copyright law (Shemtov, 2017). Such an extension of protection might seriously limit legitimate access to the software code, for instance when developing other interoperable pieces of software. Therefore, Article 6 of the Software Directive explicitly excludes the possibility to draft the licence terms prohibiting the reproduction of code *'to obtain the information necessary to achieve the interoperability of and independently created computer program with other programs'.* The CJEU in its decision in *SAS* confirmed that licence terms cannot prohibit the uses of software that do not infringe copyright. In particular, the CJEU concluded that *'to accept that the functionality of a computer program can be protected by copyright would amount to making it possible to monopolise ideas, to the detriment of technological progress and industrial development.'*[(18)]

Also, in the case of contracts of adhesion, where the negotiating power of two parties to the contract is unequal, there can be additional limitations to the contractual terms, based on public policy considerations. In the European Union, Council Directive 93/13/EEC of 5 April 1993 on unfair terms in consumer contracts protects consumers against unfair standard contract terms imposed by stronger parties. In very specific cases, involving abuse of the dominant position by the software firm, EU competition law may also be invoked to challenge licence provisions that limit the use of the software beyond the protection provided by copyright.

Licences have proved to be a useful tool not only for enhancing the appropriability but also for guaranteeing open access to the source code. Pioneers of the *copyleft* concept shaped the licence terms in such a way as to ensure that software and derivative works built upon it would be distributed without licence fees. This set in motion important developments giving rise to an array of licensing models providing alternatives to proprietary licensing.

[(18)] Judgment of 2 May 2012, *SAS Institute Inc. v World Programming, Ltd.* C-406/10. World Programming Ltd.

5. FREE AND OPEN-SOURCE SOFTWARE

Early pioneers of the OSS 'movement' found standard EULA terms to be very restrictive and encroaching upon cultural norms then prevailing among the software developer community, allowing for sharing and tinkering with the software code (William, 2002). They started to use copyright law and licensing to facilitate access to the source code. These ideas are reflected in the 'four freedoms' of the Free Software Foundation (FSF) created in 1985, which are listed below:

- the freedom to run the software for any purpose;
- the freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this;
- the freedom to redistribute copies so you can help others;
- the freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this[19].

It is important to note that the four freedoms endorsed by the FSF do not preclude the possibility to charge a price for the distribution of 'free' software or for related services. Rather, the FSF rules imply royalty-free licensing of the software.

As already discussed, writing complex software is a collaborative effort involving many people who work on pre-existing content. Copyright provisions do not prohibit a contributor to re-claim rights in works that build on the pre-existing works released into the public pool and hence open-source contributors need to restrict the possibility that ownership is claimed by downstream software contributors (Merges, 2004). Therefore, open-source relies on legal instruments governing intellectual property protection such as copyright, contract law and patent law to enable the use and modification of the source code (Walden, 2013). In many jurisdictions, copyright is automatically attached to the software, so no one can use it without explicit agreement of the author. Licences are the legal way to grant users the right to the usage of software. The original goal of open-source licences was the opposite of the original goal of IPR. While IPR is seen as a way to internalise the externalities from knowledge, open-source licences were designed to create externalities to prevent others from internalising the benefits stemming from knowledge (Merges, 2004).

In 1983, Richard Stallman started the GNU project to provide an alternative to the proprietary UNIX operating system (Haff, 2018). In 1988 the first version of the GNU general public licence (GNU GPL) was released. GPL used copyright terms to force the release of the derivative software under similar terms, guaranteeing access to the source code. GNU GPL contained the viral clause whereby every modification must be distributed under the same terms (the so-called copyleft clause). However, the viral clause embedded in GNU GPL terms may be seen by some as equally restrictive as proprietary licences and might discourage the involvement of commercial entities. As a consequence, GNU GPL may potentially limit the pace of further software innovation (Phillips, 2009). In 1988, the open-source initiative (OSI) was

[19] As listed on the web page https://www.gnu.org/philosophy/free-sw.html.

founded to promote the development of permissive licences that would be more appealing to the commercial entities. OSS licences specify to what extent OSS can be mixed with proprietary code, how it can be relicensed and if it contains special privileges for the original copyright holder (Rajala et al., 2006).

On the most general level OSS licences can be classified as:

Copyleft licences such as GNU GPL that require that any change made to the code of a program, which is distributed outside of the focal organisation, has to be made available on the same terms.

Permissive licences such as Apache, Berkeley Software Distribution (BSD), MIT. Permissive licences do not include the requirement of the distribution of the changes in the code under the same terms as the core code (hybrid OSS and proprietary).

The fundamental difference between copyleft and permissive licences lies in the fourth freedom defined by the FSF – freedom to distribute derivative software. Under a copyleft licence, the licensee may distribute modifications of the program only under the same terms. Under permissive licences, the licensee is free to distribute a derivative version of the software under any license terms, including proprietary licences (Phillips, 2009).

In fact, OSS licences can be placed on the continuum between permissive licences, standard reciprocity and strong reciprocity types (Välimäki, 2005). Firms select the optimal licence method depending on their business model as well as the usage they want to make of OSS (Hecker, 1999). In recent years, there has been a growing shift in the OSS community toward more permissive licences. In accordance with the data maintained by Black Duck by Synopsys, among the projects using one of the most popular licences, two-thirds used a permissive licence such as MIT, Apache 2.0 or BSD (Haff, 2018).

Thanks to these more liberal licensing schemes a whole ecosystem of companies emerged, whose business models rely on combining OSS with proprietary elements. This enables them to profit from their engagement in the development of OSS. In the words of Anderson (2013) *'[t]he market has decided that there's a place for all three models: totally free, free software and paid support, and good old pay for everything. (...) It's a hybrid world, with free and paid coexisting.'*

The recent growth of the popularity of OSS has been facilitated by the rapid growth of networking technologies that make cooperation within large knowledge sharing communities possible. Those technologies reduce the costs of creating and sharing knowledge among their members, while allowing for the highly decentralised character of knowledge production. In such an environment, the *free riding*[(20)] problem is no longer a crucial one due to the internet and low marginal costs. Even if there is free riding, low marginal costs mean that the producer does not lose a lot in the event that others are freeriding on the works of others. The sheer scale of some communities, enabled by the internet, is large enough to sustain the most valuable projects, even if the percentage of users actively contributing to the community with their work is small in comparison with the number of users benefiting from open-source.

(20) The free riding problem is a type of market failure where the users of a public good do not pay for their use of that good.

6. USE OF OSS LICENSES BY FOR-PROFIT FIRMS

6.1 MOTIVATION TO USE OPEN-SOURCE LICENCES BY COMMERCIAL ENTITIES

A cornerstone of OSS is the open dissemination of knowledge (source code). This system cannot be sustained without providing a mechanism rewarding contribution to the common knowledge pool. The incentive system is a crucial factor in the long-term success of any open technology system (Foray, 2006).

Open-source was promoted initially by individual developers. Individuals' incentives include own use, signalling, intrinsic (such as enjoyment) and extrinsic ('ego boost' and reputation) motivations (Mauer et al., 2006), as well as sharing knowledge, learning and acquiring new skills (Ghosh et al., 2002).

However, solitary developers are only a minor, although valuable, part of the OSS community. Flagship OSS projects are built nowadays by professional programmers working at big organisations (Young, 1999). It is therefore important to explain the motivation of for-profit entities for their engagement in the development of code that is open and can be accessed by third parties, often without additional pecuniary costs.

Initially, scholars investigating the motivation of firms were focusing on the lower costs associated with adopting OSS solutions. Adapting OSS to the particular needs of a firm often requires less engagement from in-house developers than starting a similar project from scratch. Open-source allows for easier customisation of the code to the particular needs of the company (Haff, 2018). Also, OSS software benefits from the collective effort of many developers who are able to detect potential bugs, since, as famously pointed out by Raymond (1997), '*given enough eyeballs, all bugs are shallow*'. OSS may be therefore the fastest way to market software and fix the bugs in the code.

In recent years, however, the lower cost aspect of OSS has become less important and more researchers are pointing to the fact that OSS benefits are shifting from low cost operational considerations to more strategic motives (Morgan & Finnegan, 2014).

Active involvement in open-source product development may be a strategic move to undermine the market position of the dominant market player (Merges, 2004), especially if the dominant market player owns a 'bottleneck' asset that enables it to appropriate disproportionally the returns from the innovation of others (Teece, 1986; Teece, 2018).

Given the increasing returns to adoption and network externalities, opening up technology may be a good strategic option to make software innovation into a position of dominant design (Schilling, 2017). The opening of technologies facilitates more rapid adoption and a larger user base may create an attractive market for complementary products and services. Even within the largest corporations, there is insufficient capacity to actually test and market all of the ongoing projects. Converting a project to the OSS model creates an opportunity to discover new applications, previously not envisaged by its original developers, and to hire new specialists. Investment in OSS may be also a way to ease the negative perception by others of the dominant player as it may be seen as a credible assurance that the firm will not assert excessive control over the software in the future (Merges, 2004).

A qualitative field study of the motives of 11 European firms conducted by Morgan & Finnegan (2014) revealed that, at least for some of them, engagement with open-source was initiated by the demands of their (potential) clients or by the limits of the proprietary business models they used so far. In such cases, the value creation opportunities related to OSS involvement were not initially envisaged by adopting firms and became recognised by them at a later stage (Morgan & Finnegan, 2014).

Most of the firms interviewed by Morgan & Finnegan (2014) pointed to the benefits they gained by their engagement in the OSS, but not all of them had a pecuniary character or were reflected in higher revenue streams. Most of the value captured was related to easier access to knowledge and human resources outside the firm. This resulted in higher innovation output and quicker learning and enabled firms to experiment with customisation by mixing different components of software.

The research of Morgan & Finnegan (2014) also identified some risks associated with engagement in OSS. Representatives of firms surveyed indicated that one of the biggest risks impeding their more intensive use of OSS solutions was related to a lack of centralised support and insufficient documentation due to the decentralised structure of OSS collaboration. There are also organisational risks related to changing the company's culture from focusing on closed innovation to a more open attitude. Firms were reluctant to 'giving away' their code, perceived as a valuable asset enabling value appropriation from their innovation efforts.

Additionally, firms that decide to increase their engagement within OSS may be motivated by important uncertainties regarding the scope of protection available under various types of IPR and eligible licence provisions extending the scope of copyright protection, as discussed in Section 4. As documented by Huang (2016), for those firms active in the genomics industry, in periods of uncertain IPR protection conditions, they release and acquire more knowledge through open science practices, but after the reduction in uncertainty associated with granting a patent, their strategy switches toward commercial science.

6.2 BUSINESS MODELS MIXING OSS WITH PROPRIETARY CODE

Specific features of the software, the evolution of the software sector and uncertainties regarding software protection shifted the focus of competition from the software itself to the business models enabled by it. According to the profiting from innovation (PFI) framework, the strength of the appropriability regime shapes the strategic choices of firms. In the face of weak appropriability regimes, firms rely on other value capturing mechanisms such as complementary assets or creative business models that help them benefit from their innovation (Pisano & Teece, 2007). Pisano & Teece (2007) argue that IP protection is not always the best strategy to capture value from innovation.

Although it is hard to compete with free, it is possible and there are examples of companies that built their business models on offering products or services that are better or different from the free versions (Anderson, 2013). Some business models are based on finding scarce complements to the abundant free offers, such as selling support services or customising OSS for the particular needs of a client. The key to successful business models based on a combination of open and proprietary services is to create complementarities, where the free services serve to help improve the paid part of the business (Anderson, 2013). As OS software licences focus on the developers and satisfy primarily their needs, OSS developers may underinvest in usability appealing to the average user (Phillips, 2009). This may create business opportunities for firms which offer complements that increase usability and bridge the gap between the sophisticated and the mass customer.

Firms have developed a variety of ways to include OS in their business models. They differ in the relationship with the OS community (Dahlander et al., 2005), the strength of their orientation towards OS (Bonaccorsi et al., 2006) and other elements such as the type of offering, target market, product versus service orientation, with the firm's resources and licensing model determining which elements of the offering are free and which are commercialised (Rajala et al., 2006). Firms that include OSS in their business model may provide tangible (hardware) as well as intangible products (software) in which OSS is embedded. The products they develop may use in-house development or exploit the work of the OS community. Companies may target the mass market or specific markets for customised solutions (Rajala et al., 2006; Bonaccorsi et al., 2006). Finally, firms can combine the traditional software business model and OS by using dual licences in which they offer one product but with two types of licences, one which is free for distribution and a paid licence for commercial users (Välimäki, 2003). Several studies have reported that companies tend to have several revenue sources and combine both product offerings and services (Bonaccorsi et al., 2006; Ghosh et al., 2002; Lakhani et al., 2005). Services may include software migration, installation, support, the development of ad hoc solutions, the marketing of software, consulting and training.

Table 6.1 presents eight possible business models that include open-source developed by Rajala et al. (2006).

Table 6.1 Types of business models including OSS

Revenue Model	*Description*	*Licence Types*	*Revenue Sources*
Support selling	A for-profit company provides support for software that is distributed free of charge.	Any	Revenue comes from media distribution, branding, training, consulting, custom development, and post-sales support for physical goods and services.
Loss-leader	A no-charge open-source product is used as a loss-leader for traditional commercial software, i.e. the software is made free with a hope that this will stimulate demand for a related offering the company has.	Varies	Complementary offerings, e.g. other software products.
Widget-frosting	Companies that are in business primarily to sell hardware can use this model for enabling software such as driver and interface code. By making the needed drivers open, the vendor can ensure that they are debugged and kept up to date.	Any	The company's main business is hardware. This is quite similar to the loss-leader model.
Accessorising	Companies which distribute books, computer hardware and other physical items associated with and supportive of OSS.	Any	Supplementary offerings.
Service enable	OSS is created and distributed primarily to support access to generating revenue from consulting services and online services.	Any	Service fees.
Brand licensing	A company charges other companies for the right to use its brand names and trade marks in creating derivative products.	Strong reciprocity	Copyright compensations.
Sell it free it	A company's software products start out their product life cycle as traditional commercial products and then are converted to open-source products when appropriate.	Alteration of licence type	Initial revenue from software product offerings converted into other models, e.g. the loss-leader model.
Software franchising	A combination of several of the preceding models (in particular 'Brand licensing' and 'Support selling') in which a company authorises others to use its brand names and trade marks in creating associated organisations doing custom software development, in particular geographic areas or vertical markets.	Strong reciprocity	The franchiser supplies franchisees with training and related services in exchange for franchising fees of some sort.

Source: Rajala et al. (2006).

7. DATA AND METHODOLOGY

To verify the scale of OSS engagement and the importance of OSS for European firms active in the software industry, a survey was conducted between 6 August 2018 and 8 November 2018 (the survey questions are reproduced in the Annex). The survey was administered to firms active in the following sectors, as defined by Eurostat's NACE classification:

- 58.2 Software publishing;
- 62 Computer programming, consultancy and related activities;
- 63.1 Data processing, hosting and related activities; web portals.

The survey collected valid responses from 1 364 firms, active in the software sector and located in one of the following countries: Austria, Denmark, Estonia, Finland, France, Germany, Italy, Poland, Romania and the United Kingdom[(21)].

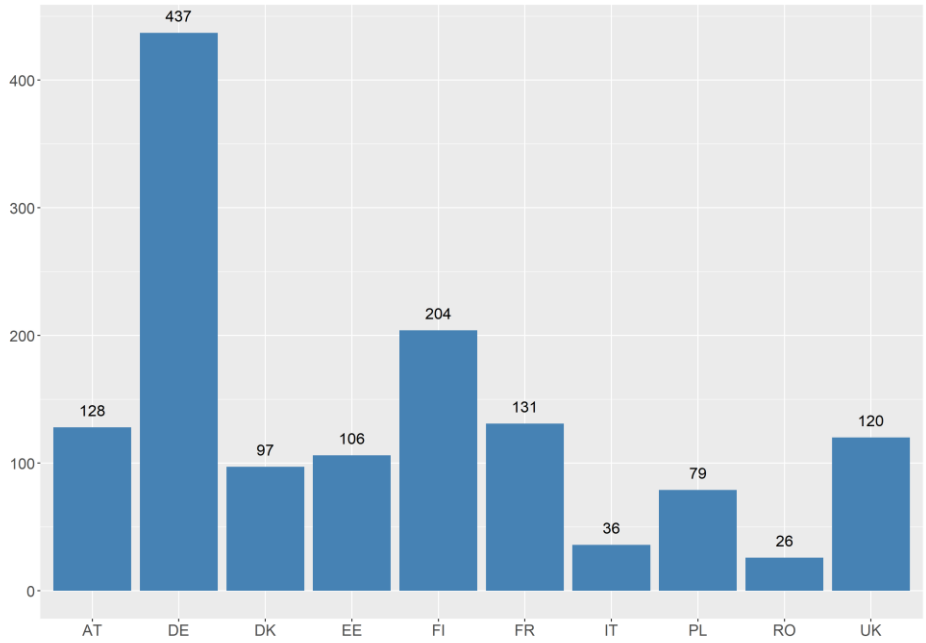
Firms active in the software sector are not the only ones affected by OSS. Software is today a general purpose technology, and firms from virtually all sectors rely on computer programs, including programs licensed under OS terms. According to the existing statistics for almost 80 % of CII-related patent filings, applicants came from the manufacturing sectors, with the highest share attributed to the manufacturing of computer, electronic and optical products (Frietsch et al., 2015). However, whereas in other sectors only a minority of firms have the experience and necessary knowledge related to OSS, in the three software sectors listed above every firm would be able to answer the survey questions.

Figure 7.1 presents the distribution of respondents among the countries covered by the survey, Figure 7.2 among subsectors of IT industries and Thus, the responses of the firms which are overrepresented in the sample (in comparison with the Eurostat statistics) are weighted lower, while the responses of the firms which are underrepresented in the sample are weighted higher in the aggregate statistics.

Figure 7.3 among size categories.

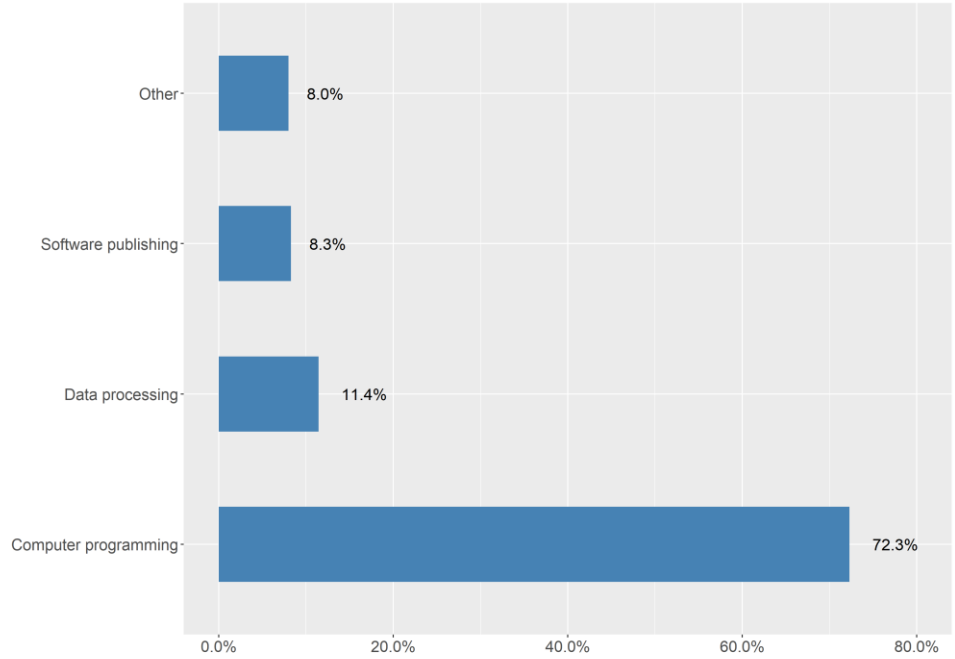
[(21)] To reduce the burden on firms and to reduce the project costs, the survey was conducted in 10 EU Member States: AT, DE, DK, EE, FI, FR, IT, PL, RO, UK. The geographical and socio-economic diversity of the EU has been preserved. At the time of the survey, in late 2018, the United Kingdom was still a member of the EU and hence UK-based companies are included in the study.

Figure 7.1 Number of respondents by country



Note: Based on firms' information available in ORBIS Bureau Van Dijk (N= 1 364).

Figure 7.2 Number of respondents by sector



Note: Based on Question 3: 'In which sector is the main activity of your business?' Single choice. N = 1 364.

Computer programming is the sector represented by the highest number of respondents, corresponding to 72.3 % of the firms surveyed. Firms active in data processing constitute 11.4 %; software publishing 8.3 % and firms active in other sectors – 8 % of respondents.

Micro firms are the group with the biggest number of respondents and constitute over 70 % of the sample. Small firms' share amounts to 21.6 % of the sample, medium firms make up 5.6 % and large firms 1.4 % of the sample[(22)].

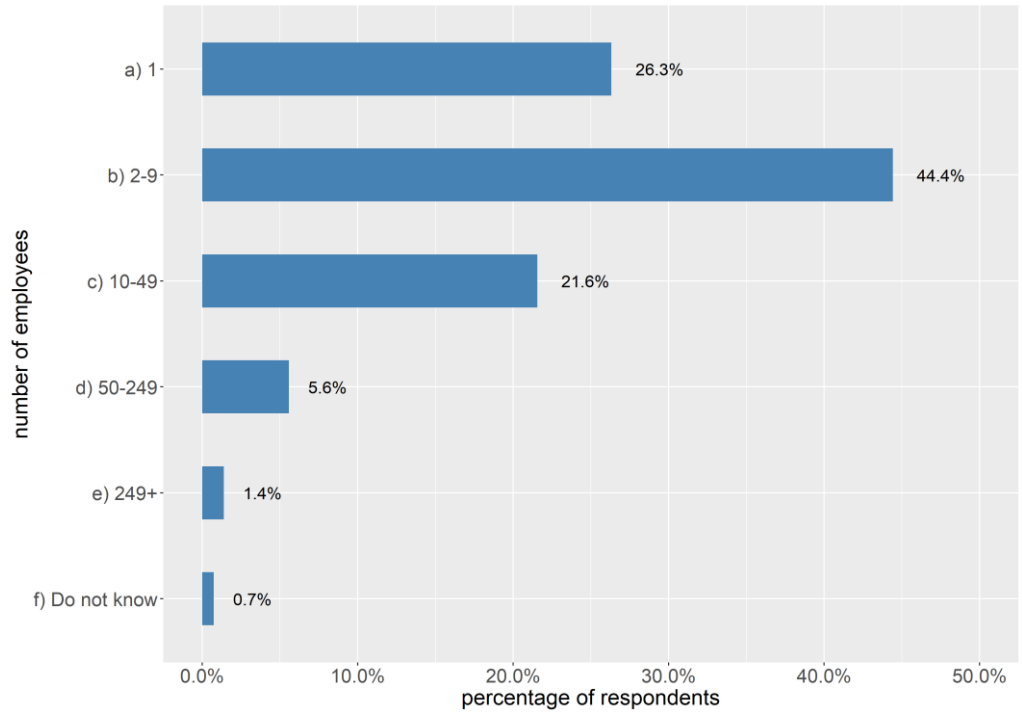
The sample has been compared with the composition of the firms by size and sector in the Eurostat data for the sector[(23)]. This analysis showed that the composition of the sample does not match exactly the composition of the sector. Bigger firms are overrepresented and there is an imbalance between industries represented in the sample as compared with the Eurostat statistics. To mitigate this problem, all the statistics presented in the following section reflect weighted data with individual weights taking into account a firm's size and the sector it represents. Thus, the responses of the firms which are overrepresented in the sample (in comparison with the Eurostat statistics) are

(22) The classification of firms by size follows the method used by the European Commission. Micro firms are those with 10 employees or less; small firms have between 11 and 50 employees; medium-sized firms have between 51 and 250 employees. Additional criteria related to annual turnover and the size of the firm's balance sheet also apply.

(23) For that comparison data from table sbs_sc_1b_se_r2 *Services by employment size class*.

weighted lower, while the responses of the firms which are underrepresented in the sample are weighted higher in the aggregate statistics.

Figure 7.3 Number of respondents by firm size



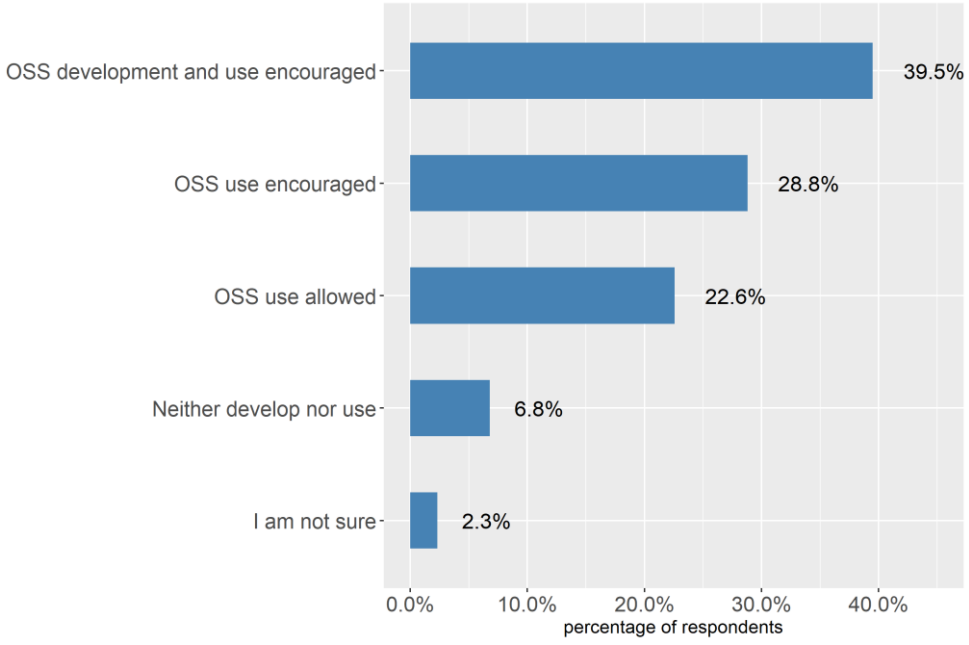
Note: Based on Question 1: 'What was the approximate number of employees in your organisation at the end of 2017?' Single choice. N= 1 364.

8. DESCRIPTIVE ANALYSIS OF THE SURVEY RESPONSES

8.1 FIRMS' POLICIES TOWARD OSS

Figure 8.1 describes firms' choices regarding policy options towards the development and use of OSS applications. The biggest group of firms that took part in the survey (almost 40 %) actively encourages involvement of its employees in OSS development and use during office hours and as part of their duties. The second most popular option is encouragement to use the OS products in the firm's activity. Almost 29 % of respondents actively encourage use of open-source software but their employees are not involved in OSS development during office hours. Another 22.6 % of firms allow their employees to use OSS if it is the most appropriate tool for the task. However, their employees are not involved in the development of such software during office hours. Only 6.8 % of firms stated that their employees neither develop nor use OS products at work. Finally, 2.3 % of respondents indicated that they are not sure about their firm's policy towards OSS.

Figure 8.1 Firms' policy towards open-source

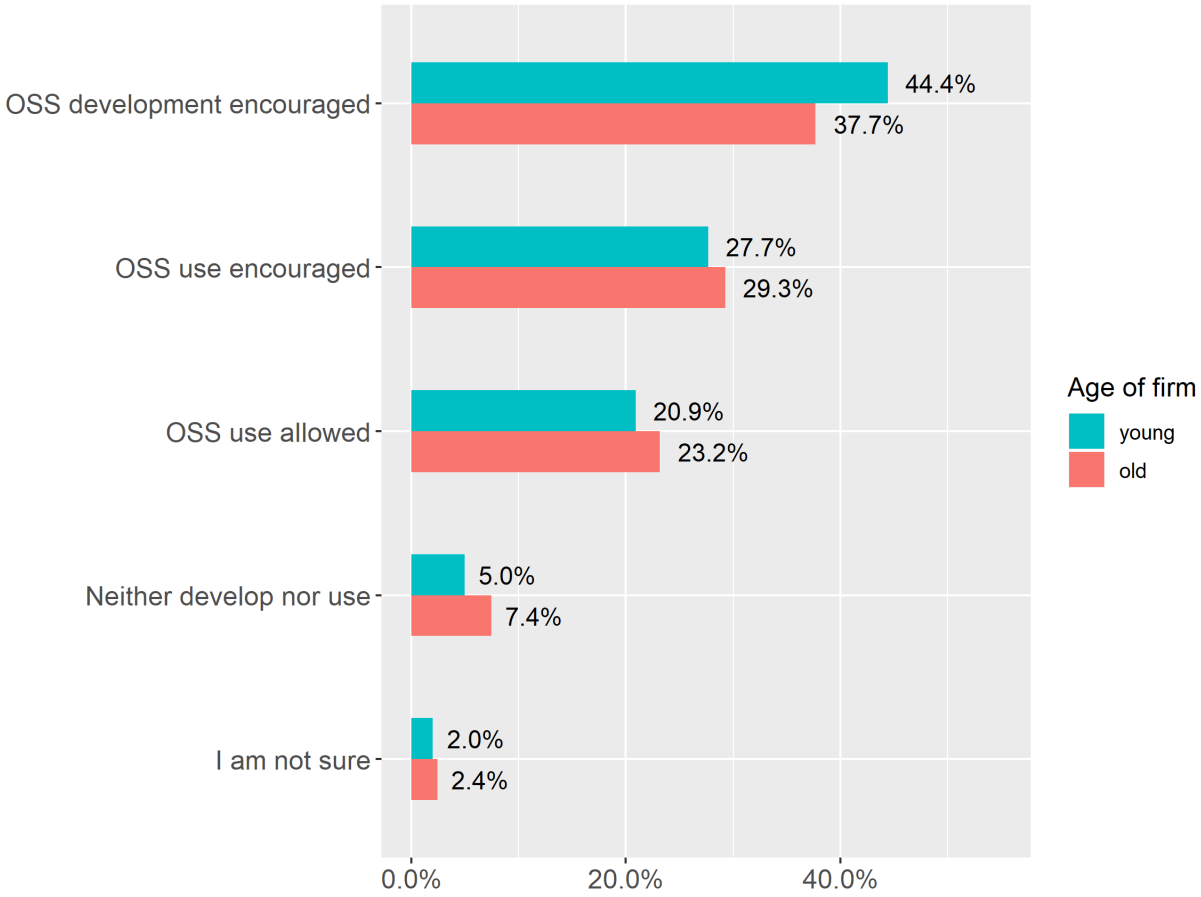


Note: Based on Question 6: 'What statements best describe your firm's policy on the development and use of open source software applications?' Single choice. N = 1 364.

As illustrated in Figure 8.2, the propensity to engage in the development of OSS is higher among younger firms (5 years' old and younger) than among older ones. Among the young firms, the share of firms actively encouraging their staff to engage in the development and use of OSS during office hours amounts to 44.4 % and is 6.7 percentage points higher than among the older firms. This difference is statistically significant on the 95 % confidence level.

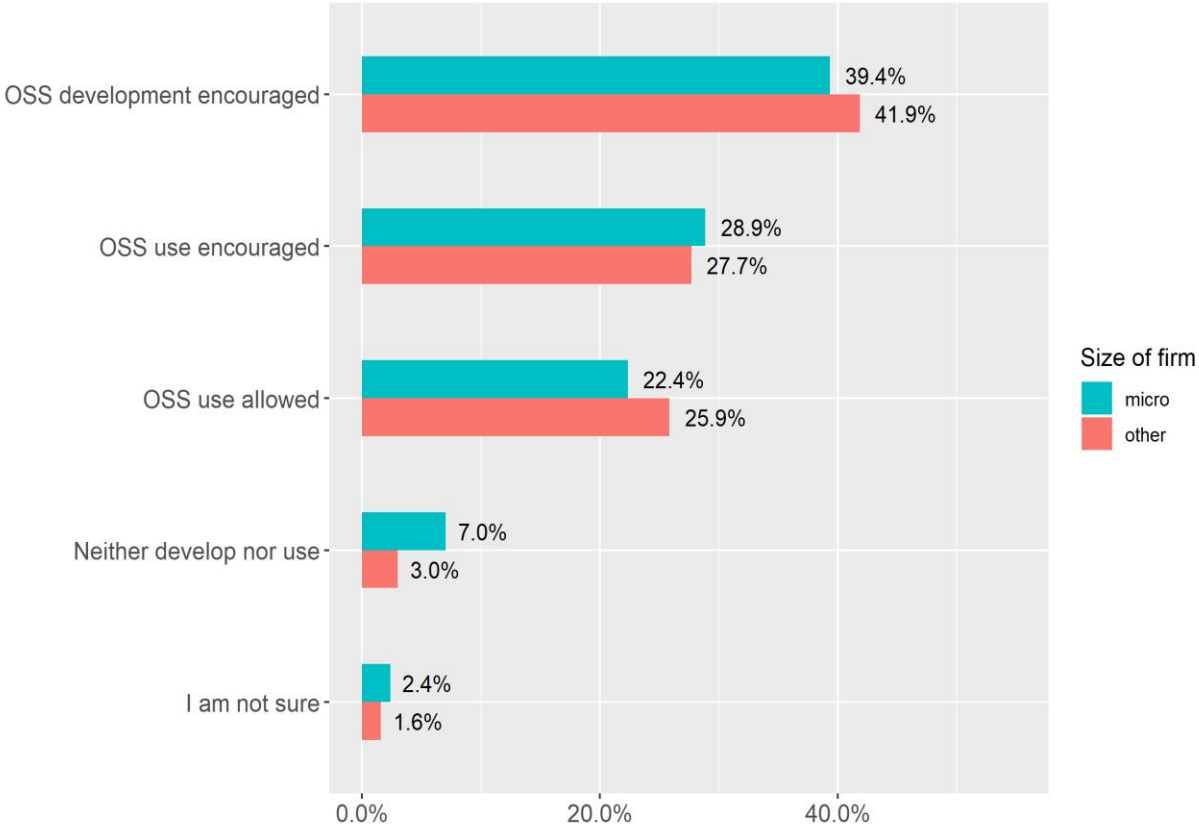
However, Figure 8.3 suggests that size of firms is not correlated with their propensity to engage in active development or use of OS products. Although there are some differences between micro and other firms in this respect, they are not statistically significant.

Figure 8.2 Age and firms' policy towards open-source



Note: Based on Question 6: 'What statements best describe your firm's policy on the development and use of open source software applications?' Young firms defined as those with their year of establishment after 2012.
N=1 363, N old = 1 044, N young = 319.

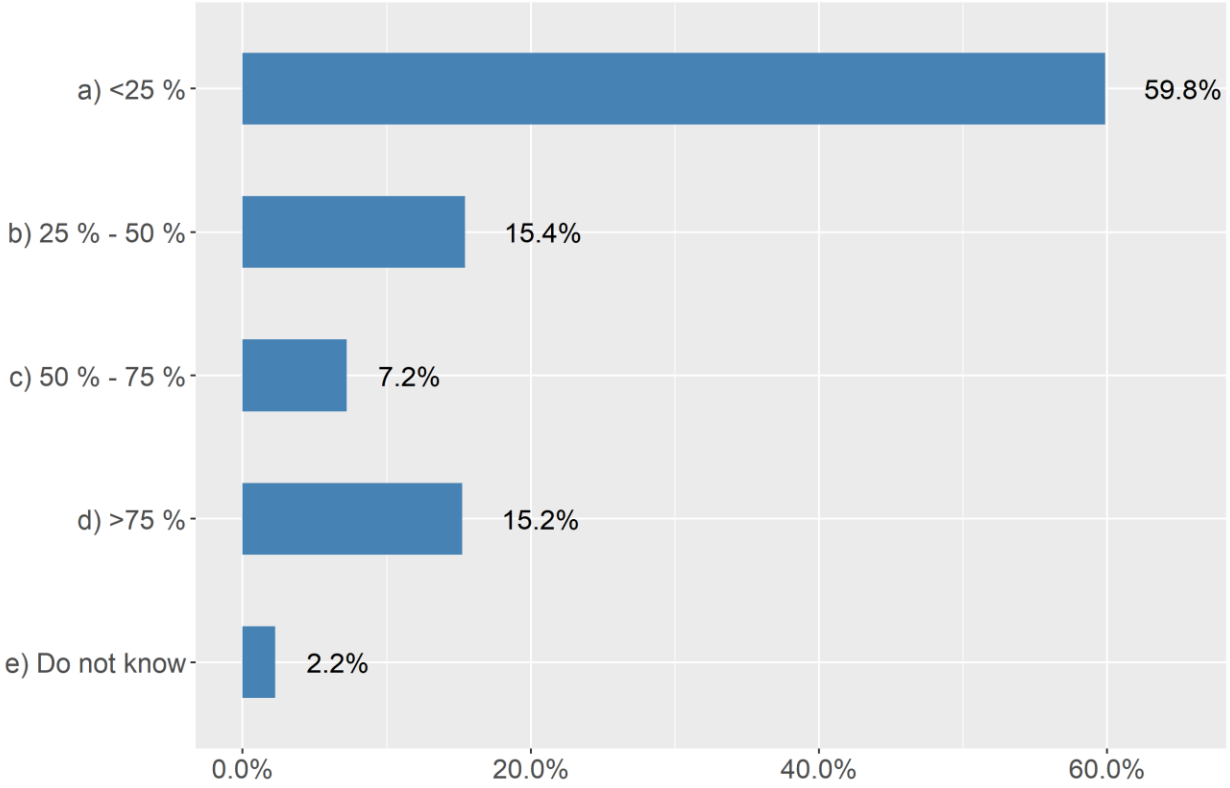
Figure 8.3 Size and firms' policy towards open-source



Note: Based on Question 6: 'What statements best describe your firm's policy on the development and use of open source software applications?'
Micro firms defined as those that comply with both employment and turnover thresholds for micro firms as defined by the European Commission in the Commission Recommendation of 6 May 2003.
N = 1 364, N micro = 972, N other = 392.

Figure 6.4 shows the percentage of working time devoted to working on OSS projects among the firms that actively encourage their employees to work on such projects during office hours. In 15.2 % of firms, time devoted to this activity is over 75 % of the overall working time of employees. In most firms (almost 60 %), the working time of employees devoted to OS software development is below 25 %. In 22.6 % of firms encouraging employees to develop OS software, employees spend between 25 % and 75 % of their time on OSS projects.

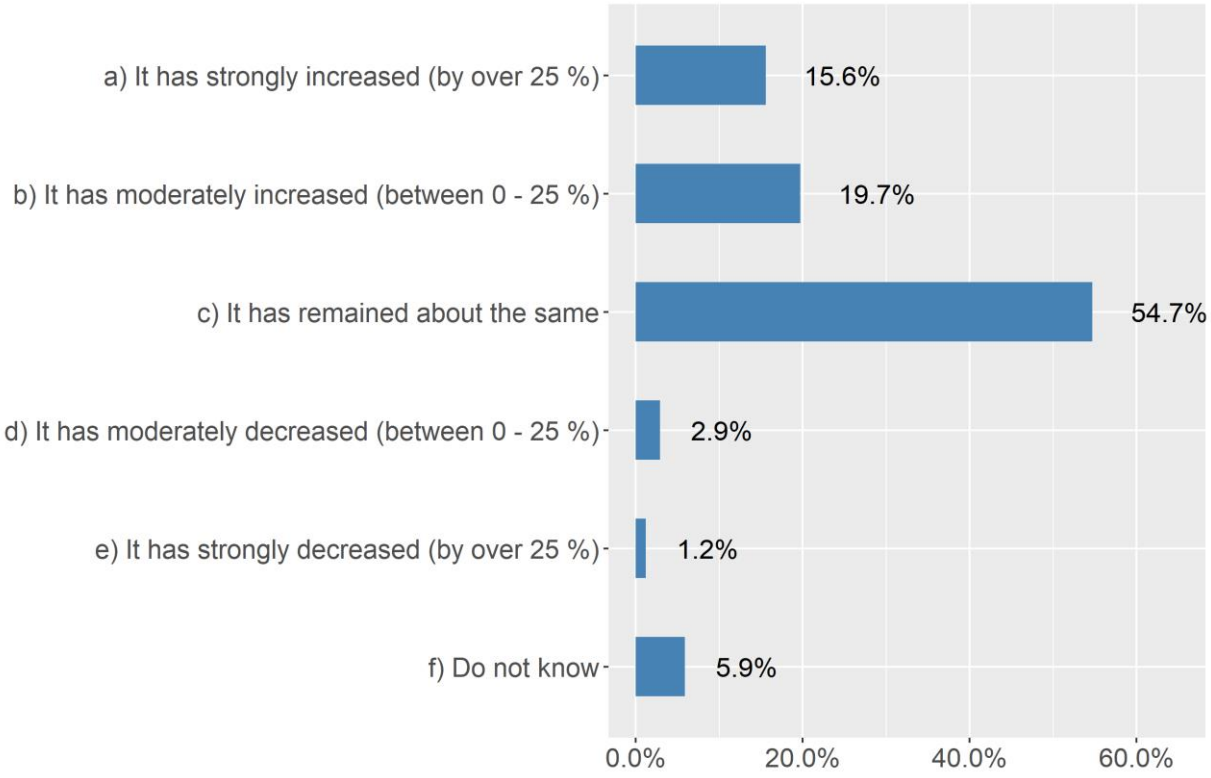
Figure 8.4 Time spent by employees on open-source software projects



Note: Based on Question 7: 'What amount of working time do employees spend on the development of open-source projects?' Single choice. Only firms that chose option a) in Question 6. N = 542.

As seen in Figure 8.5, in most of these firms (almost 55 %) the time spent on the development of OSS remained stable during the three years preceding the survey. However, firms that experienced changes in the time spent by their employees in the development of OS software reported an increase rather than a decline. More than 35 % of firms engaged in OSS development increased their involvement in OSS either moderately or strongly. Only in a small number of firms (about 4 %) did employee time spent on OSS development decrease during the three years preceding the survey.

Figure 8.5 Changes in time spent in the development of OS software (last 3 years)



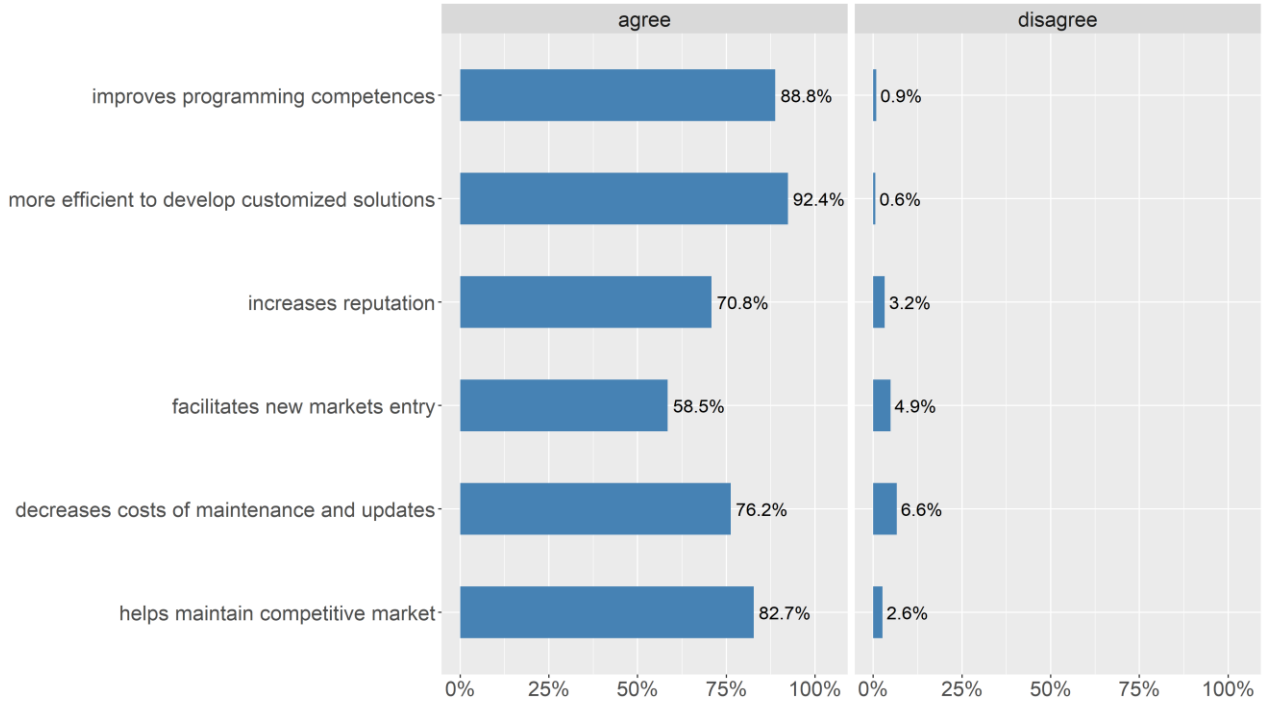
Note: Based on Question 8: 'How much has working time spent on the development of open-source software changed over the last 3 years?' Only firms that chose option a) in Question 6. N = 542.

Those 542 firms that encourage employee involvement in OSS were then asked about the costs and benefits of such involvement. The results are shown in Figure 8.6. In the opinion of the respondents, the strategic benefits from their employees' engagement in OSS outweigh the costs. More than 92 % of those surveyed agree that OSS helps to develop tailor-made solutions in a more efficient manner than proprietary alternatives. Almost 89 % agree that involvement in OSS helps to improve the overall programming competences of their employees. For 83 % of respondents OSS engagement is important because it helps to maintain a competitive and level-playing field due to standardisation and common platform building.

About 76 % of respondents agree that OSS decreases the cost of software maintenance and updates, although 6.6 % did not agree with that statement, the highest such proportion answering negatively to any of the options.

Seventy-one per cent of respondents agree that their involvement in the OSS helps them to increase reputation. Finally, 58.5 % agree that OSS facilitates their entry into new markets, with 5 % disagreeing with that statement. Thus, market entry seems to be the weakest of the motives behind the companies' involvement in OSS.

Figure 8.6 Main benefits from employees' engagement in developing open-source software



Note: Based on Question 9: 'What are the most important benefits of employee engagement in developing open-source software during office hours?' Single choice for each potential benefit. Only firms that chose option a) in Question 6.
Answers 'Strongly agree' and 'Tend to agree' have been combined into one factor 'Agree'; answers 'Disagree' and 'Strongly disagree' have been combined into one factor 'Disagree'. The figure does not show other options available to respondents: 'Neither agree nor disagree' and 'Do not have an opinion'.
N = 542.

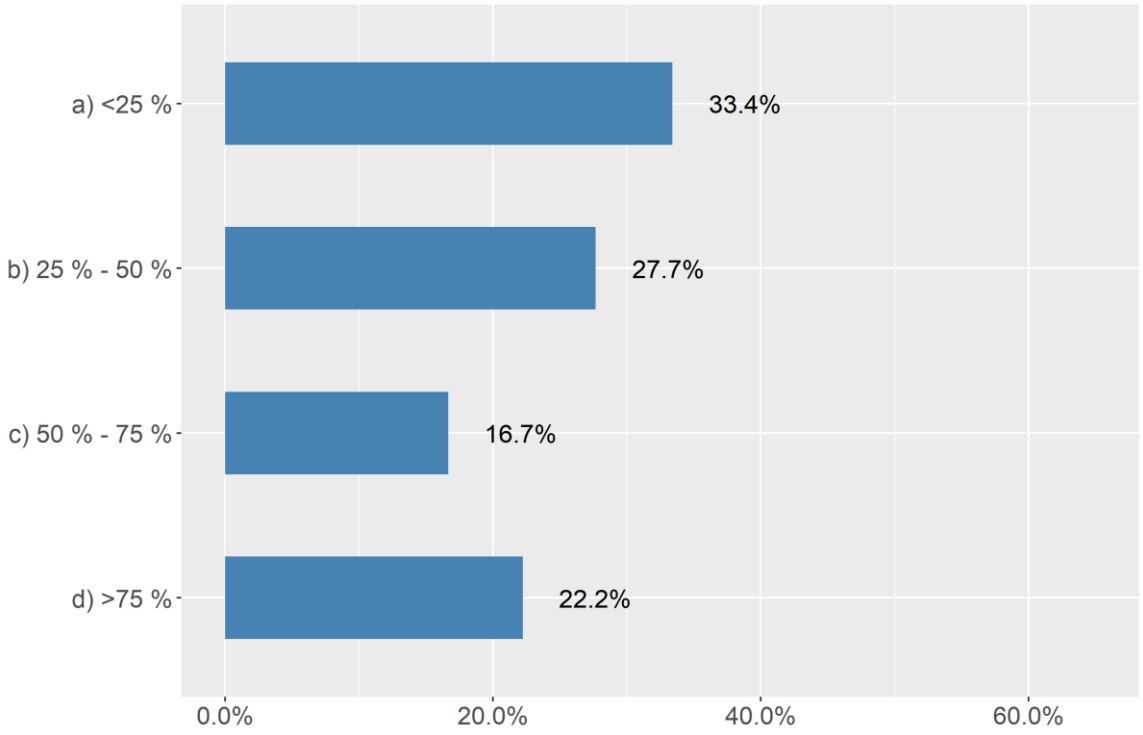
Figure 8.7 indicates that the intensity of use is higher than the intensity of development of OSS software. In 39 % of firms that encourage or allow their employees to use OSS software during office hours, the use of OSS software is higher or on a par with proprietary software. 27.7 % of firms use OSS software between 25-50 % as often as proprietary software and in 33.4 % of firms it is below 25 %.

However, as illustrated in

Figure 8.8, similarly to firms that declared engagement in OSS development, OSS users tend to maintain the intensity of use of OSS at the same level over the last three years. In almost 60 % of OSS-using firms, the level of usage remained the same during the last 3 years preceding the survey. In cases where this intensity changed, the change tended to be in the direction of greater

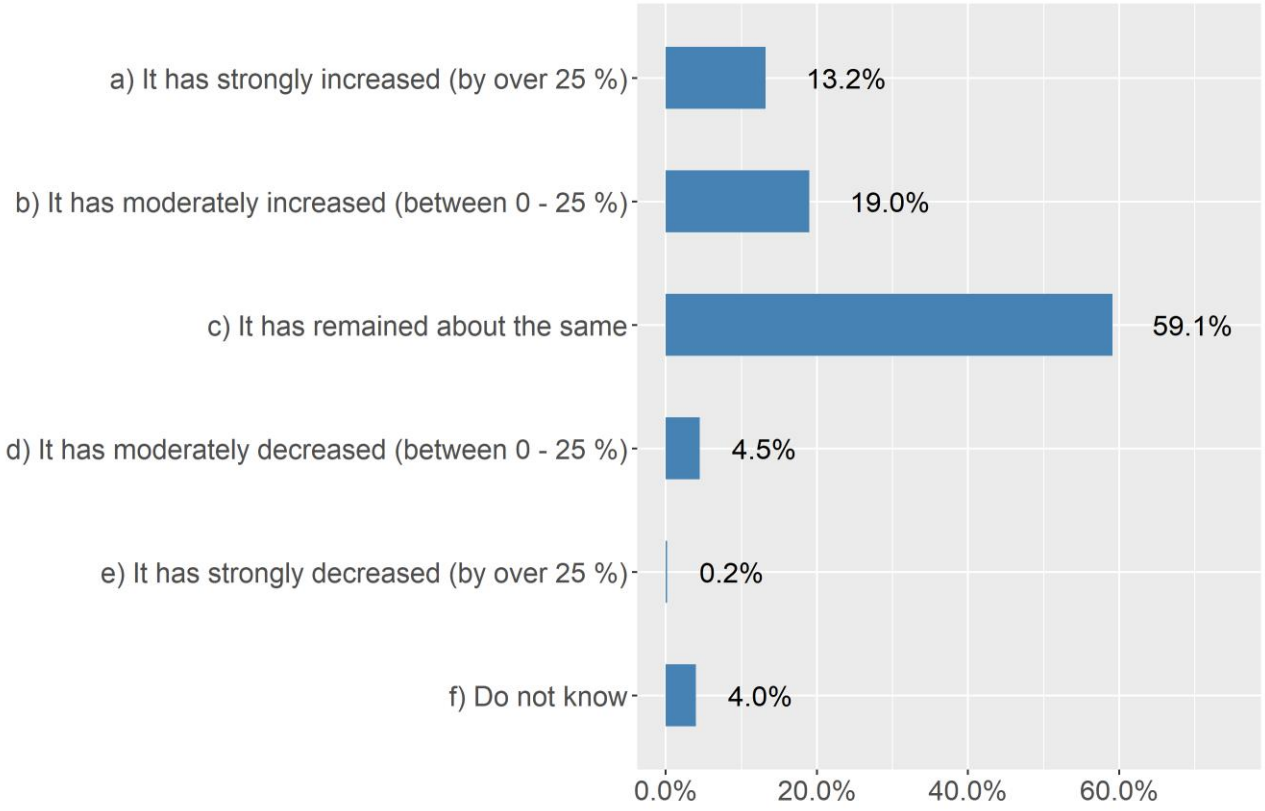
use of OSS, with 33 % of respondents reporting an increase and less than 5 % a decrease in such use.

Figure 8.7 Intensity of use of OSS



Note: Based on Question 10: 'How much open-source software is used in comparison to proprietary software in your company?' Single choice. Only firms that chose options a), b) or c) in Question 6. N = 712.

Figure 8.8 Changes in the intensity of use of OS software (last 3 years)



Note: Based on Question 11: 'How much has the usage of open-source software in your company changed over the last three years?' Single choice. Only firms that chose options a), b) or c) in Question 6. N = 712.

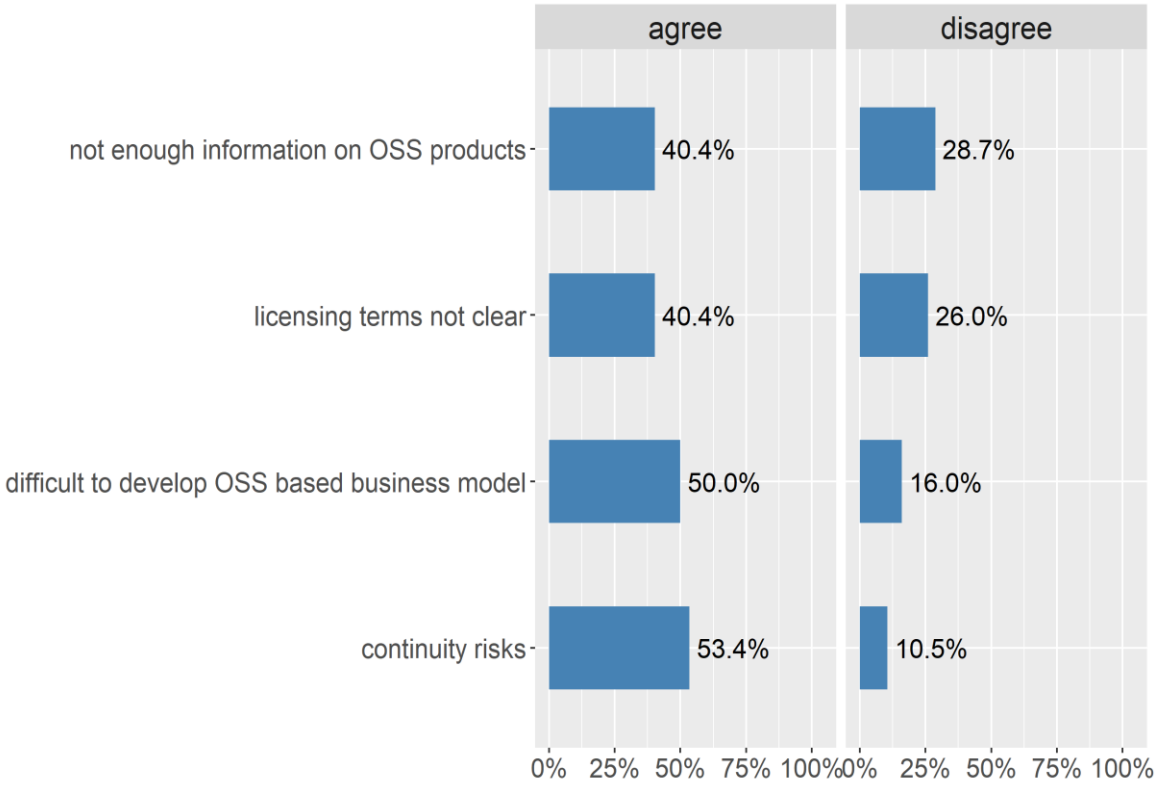
Firms that indicated that their employees were not involved in developing or using OSS were asked about the motives for such a policy choice. The answers to that question indicate that the major reason for companies not developing or using OSS is the governance model of OSS that, in their view, does not guarantee the continuity of product development and consequently may threaten the business model in the future. In the literature there are some indications that such views are not entirely unfounded. Almost no OSS licences offer warranties or envisage indemnification for users in the event of lawsuits related to the potential infringement of copyright (Phillips, 2009). There have been some prominent cases in the US in which software firms were claiming ownership of UNIX code and sued or threatened to sue IBM and other distributors of Linux[24]. There have

[24] SCO Group, Inc. v IBM, 2005 U.S. Dist.; SCO Group, Inc. v Novell, Inc., 2008 U.S. Dist.

also been some controversies regarding the possibility of developers revoking their consent for use of software development to which they had contributed (Phillips, 2009).

Thus, 53.4 % of respondents agreed with the statement that the OSS model does not guarantee continuity. Slightly lower proportions of respondents agreed with the statement that it is difficult to develop sustainable business models based on OSS. Equal shares (40.4 %) of firms that are not actively developing or using OSS products agreed that they do not have enough information on OSS products and that the licensing terms are not clear. These two statements are also ones on which many firms disagree, as more than a quarter of firms that are not engaged in OSS disagreed that the lack of information or lack of clarity of licensing were major obstacles.

Figure 8.9 Main limitations of use and development of OSS products



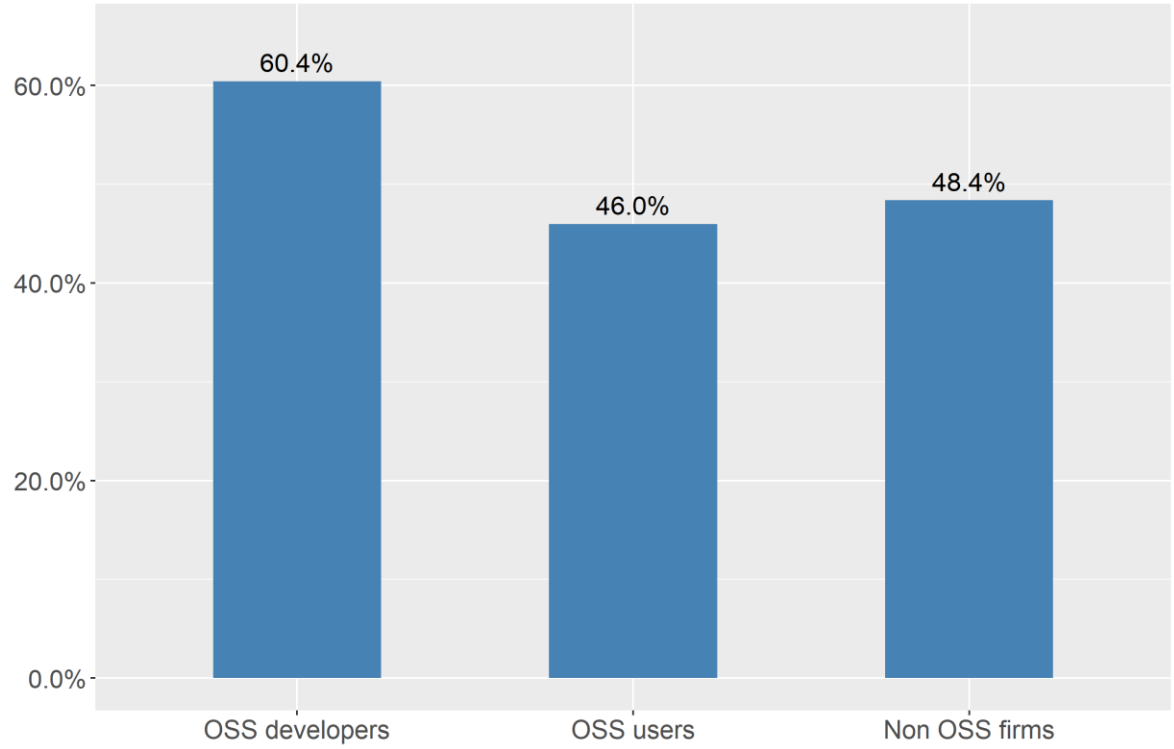
Note: Based on Question 12: 'What are the factors limiting your use and/or engagement in the development of open-source products?' Single choice for each potential constraint. Only firms that chose option d) in Question 6. Answers 'Strongly agree' and 'Tend to agree' have been combined into one factor 'Agree'; answers 'Disagree' and 'Strongly disagree' have been combined into one factor 'Disagree'. Figure does not show other options available to respondents: 'Neither agree nor disagree' and 'Do not have an opinion'. N = 79.

8.2 BUSINESS MODELS INCLUDING OSS

Given the relatively high share of firms actively encouraging their employees to develop or use OSS products, it is relevant to analyse the business models of those firms that allow them to profit from OSS activity.

A little more than half (52 %) of the firms in the sample distribute software developed by their employees. However, the share of firms distributing software is much higher among the OSS developers than among firms forming two other groups, OSS users and non-OSS firms.

Figure 8.10 Share of firms distributing software within different groups of firms



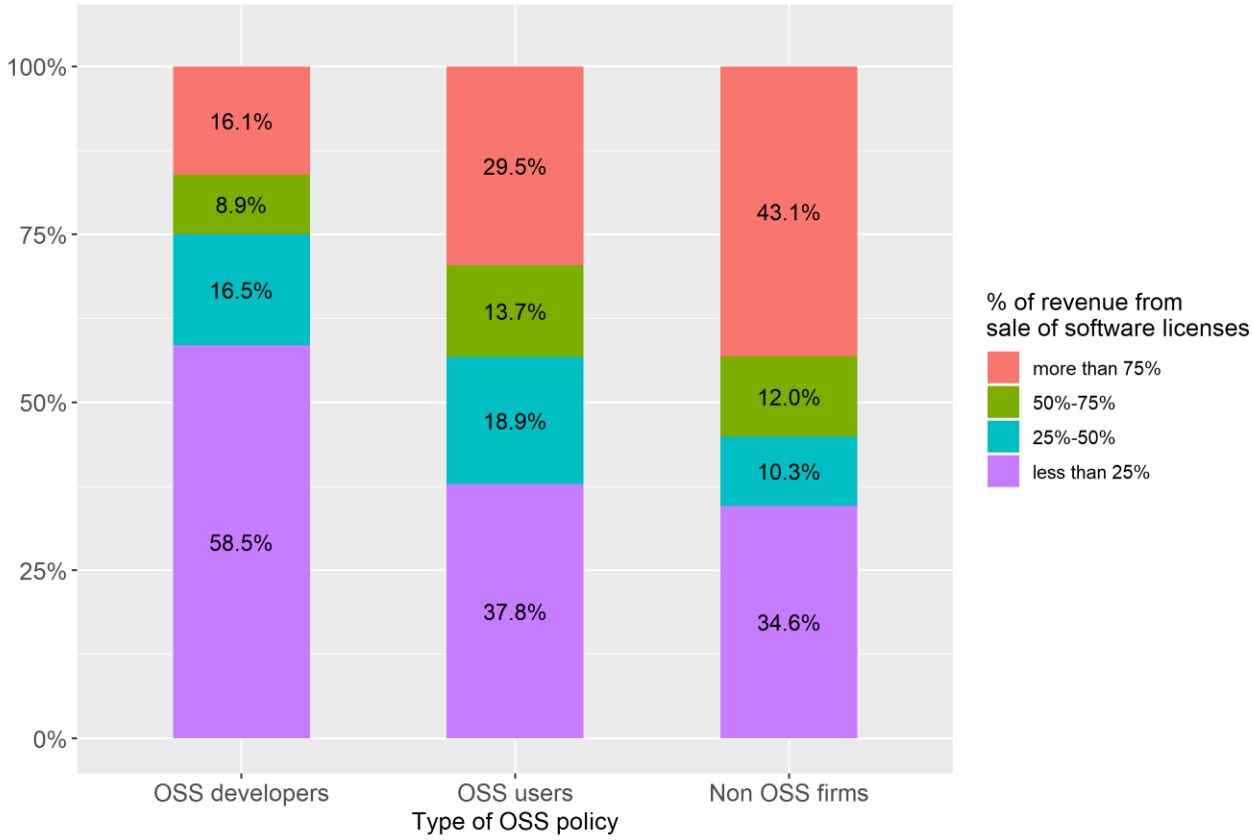
Note: Based on Question 17: 'Does your company distribute software developed by your employees?' Single choice.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a); OSS users' firms that chose options b) and c); non-OSS firms – other firms choosing options d) and e).
N = 1 364; N OSS developers = 542; N OSS users = 712; N Non-OSS firms = 110.

As shown in Figure 8.10, slightly more than 60 % of firms engaged in OSS development distribute software developed by their employees. In the other two groups of firms, the share of software

distributors is below 50 %. The difference between OSS developers and non-OSS firms is statistically significant at the 95 % confidence level.

There are large differences between firms as regards the share of revenue stemming from the sale of software licences. As shown in Figure 8.11, only 16 % of OSS developers derive more than 75 % of their revenue from the sale of software licences. In the group of firms that only use OSS, this percentage is 30 %, while for firms that neither develop nor use OS software it amounts to 43 %. For most OSS developers, the sale of software licences constitutes less than 25 % of their revenue. In other words, firms that focus on proprietary software rather than on OSS tend to derive a greater proportion of their revenue from licensing.

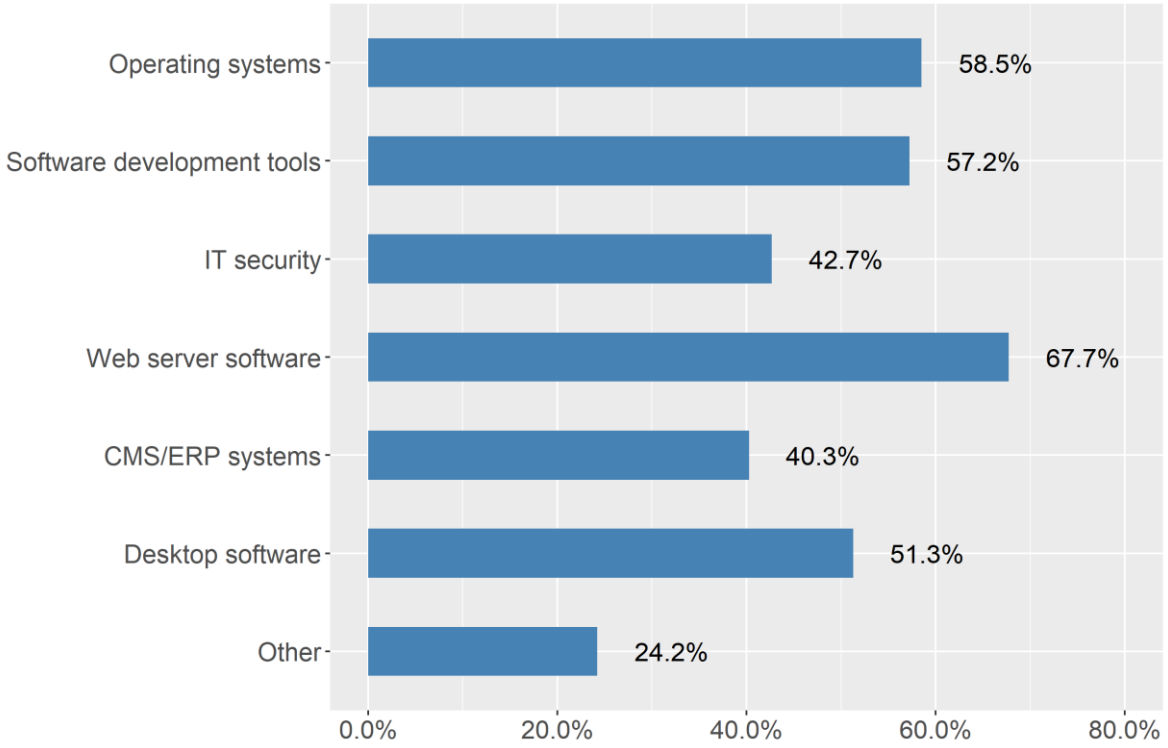
Figure 8.11 Sale of licences as a percentage of revenue of firms in the sample



Note: Based on Question 18: If answer to Question 17 is 'Yes'. 'What is the percentage of revenue derived from the sale of licences for software developed in your company?' Single choice. Firms that chose option 'Yes' for Question 17.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a); OSS users' firms that chose options b) and c); non-OSS firms – other firms choosing options d) and e).
N = 760, N OSS developers = 343; N OSS users = 363; N Non-OSS firms = 54.

As illustrated in Figure 8.12, the most popular type of OS software among the firms that use or develop OSS is web server software. Other types of frequently used or developed OSS include operating systems, software development tools and desktop software such as text processing or spreadsheets. Less common are OS IT security software and OS CMS/ERP systems[25].

Figure 8.12 Types of OSS software developed or used

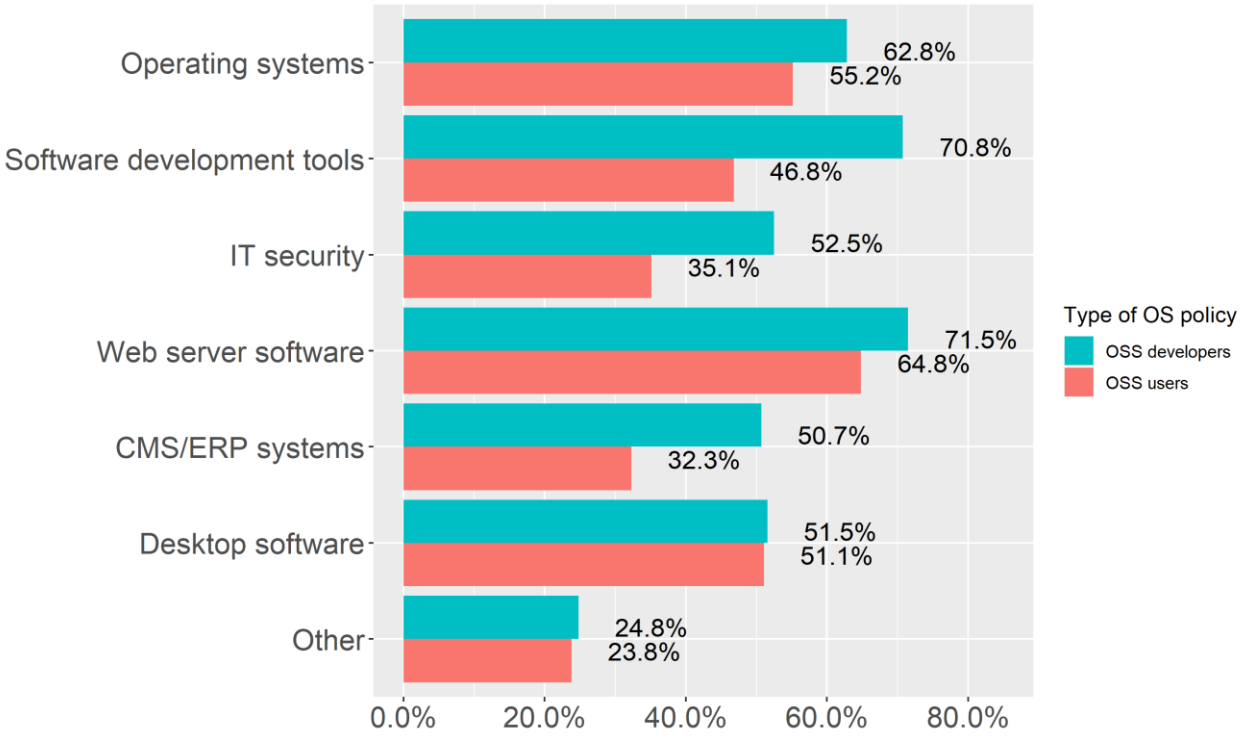


Note: Based on Question 15: 'What types of open-source applications are used/developed in your company?' Multiple choice. Only firms that chose options a), b) or c) in Question 6. N = 1 254.

Figure 8.13 breaks down this information by type of involvement with OSS: developer/user. OSS-developing firms in general engage in a broad variety of software development projects. More than 70 % of those firms develop or use web server software and software development tools. Over 60 % develop OS operating systems. More than half of the OSS-developing companies participate in development or use OS IT security software, OS desktop software and OS CMS/ERP systems. For each type of OSS software, the rate of engagement of OSS user firms is lower than that of OSS-developing firms.

[25] CMS stands for content management systems and ERP for enterprise resource planning systems.

Figure 8.13 Types of OSS software developed or used

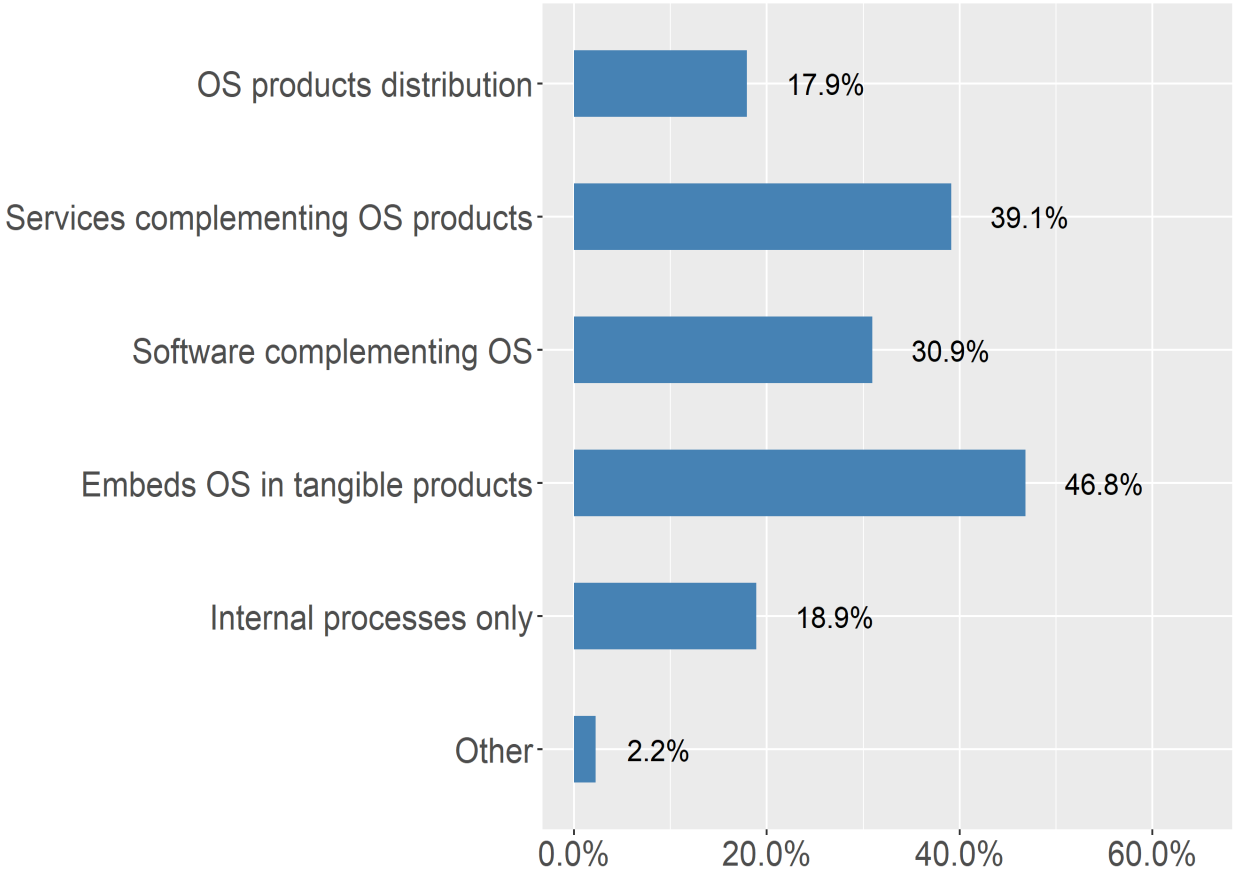


Note: Based on Question 15: 'What types of open-source applications are used/developed in your company?' Multiple choice. Only firms that chose options a), b) or c) in Question 6.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c). N = 1 254.

As discussed above, most OSS-developing firms derive less than 25 % of their revenue from the sale of software licences. OSS products are developed mainly by and for developers. As a consequence, user functionalities may be less developed in comparison to proprietary alternatives. OSS might also be considered as needing additional expertise or be less safe than proprietary alternatives. This opens business opportunities for firms providing complementary products or services mitigating those gaps and creates a symbiotic relationship between OS software developers and other types of firms (Lerner & Tirole, 2005). The following two figures analyse how OSS development or usage is combined with other sources of revenue to allow those firms to profit from their engagement in OSS.

As seen in Figure 8.14, almost half of the companies that take part in the development of OSS or use OSS products benefit from OSS by embedding OSS in tangible products (46.8 %). The second most popular option is providing services complementing OSS products (39.1 %). Almost 31 % of firms develop software complementing OSS. Relatively less popular is OSS product distribution. Almost 19 % of firms uses OSS for internal processes only.

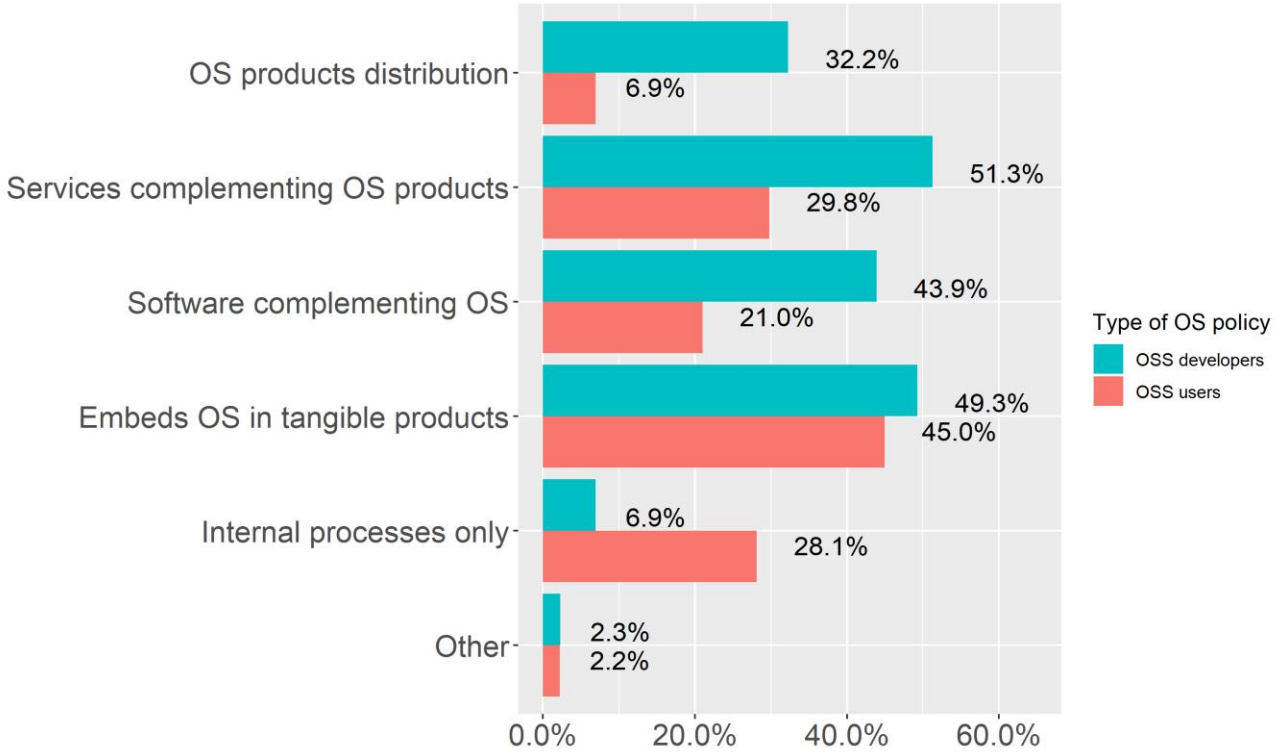
Figure 8.14 Business models combined with OSS



Note: Based on Question 13: 'How does your company benefit from involvement in open-source projects?' Multiple options are possible. Only firms that chose options a), b) or c) in Question 6. N = 1 254.

Figure 8.15 shows that OSS developers are more likely than OSS users to employ the various ways of extracting value from their involvement in OSS. One-third of developers is involved in OSS distribution as compared to 7 % in the case of OSS users. However, other ways of extracting value from OSS involvement are more common than OSS products distribution among OSS developers. More than 50 % of those firms are selling services complementing OS products, almost the same share of firms embeds OSS in tangible products and almost 44 % sells software complementing OSS.

Figure 8.15 Business models combined with OSS variations depending on OSS policy

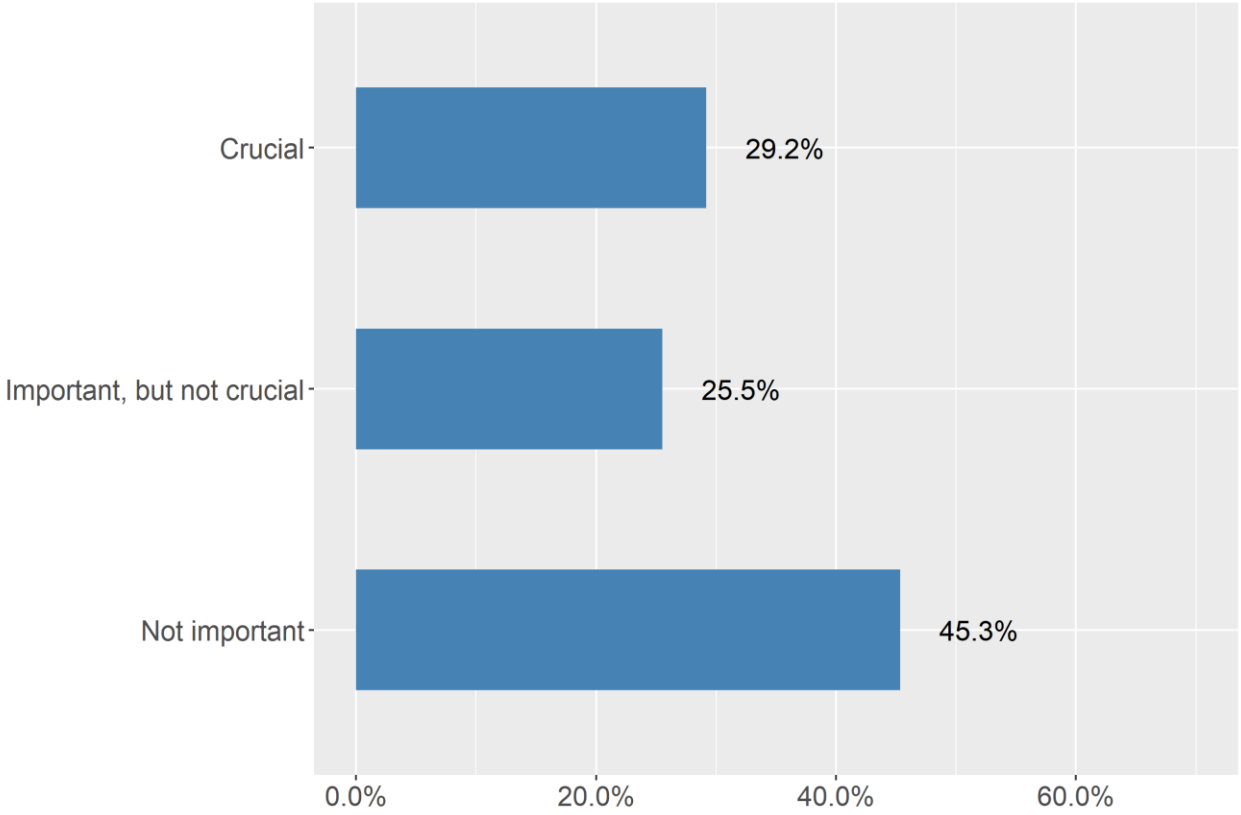


Note: Based on Question 13: 'How does your company benefit from involvement in open-source projects?' Multiple options are possible. Only firms that chose options a), b) or c) in Question 6. Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c).
N = 1 254, N OSS developers = 542; N OSS users = 712.

The existence of OSS can play a role in new business formation as well. As illustrated in Figure 8.16, the availability of OSS has been crucial or important for the decision to start the company for more than half of the firms engaged in OSS development or using OSS products in their daily work.

OSS may be more important for younger firms that started their business activity recently than for firms that have been active on the market for a longer period of time. To check whether this is the case, a separate analysis was conducted for young firms, defined as firms founded within the past 5 years, and old firms, defined as firms founded more than 5 years ago. The results of this analysis are presented in Figure 8.17.

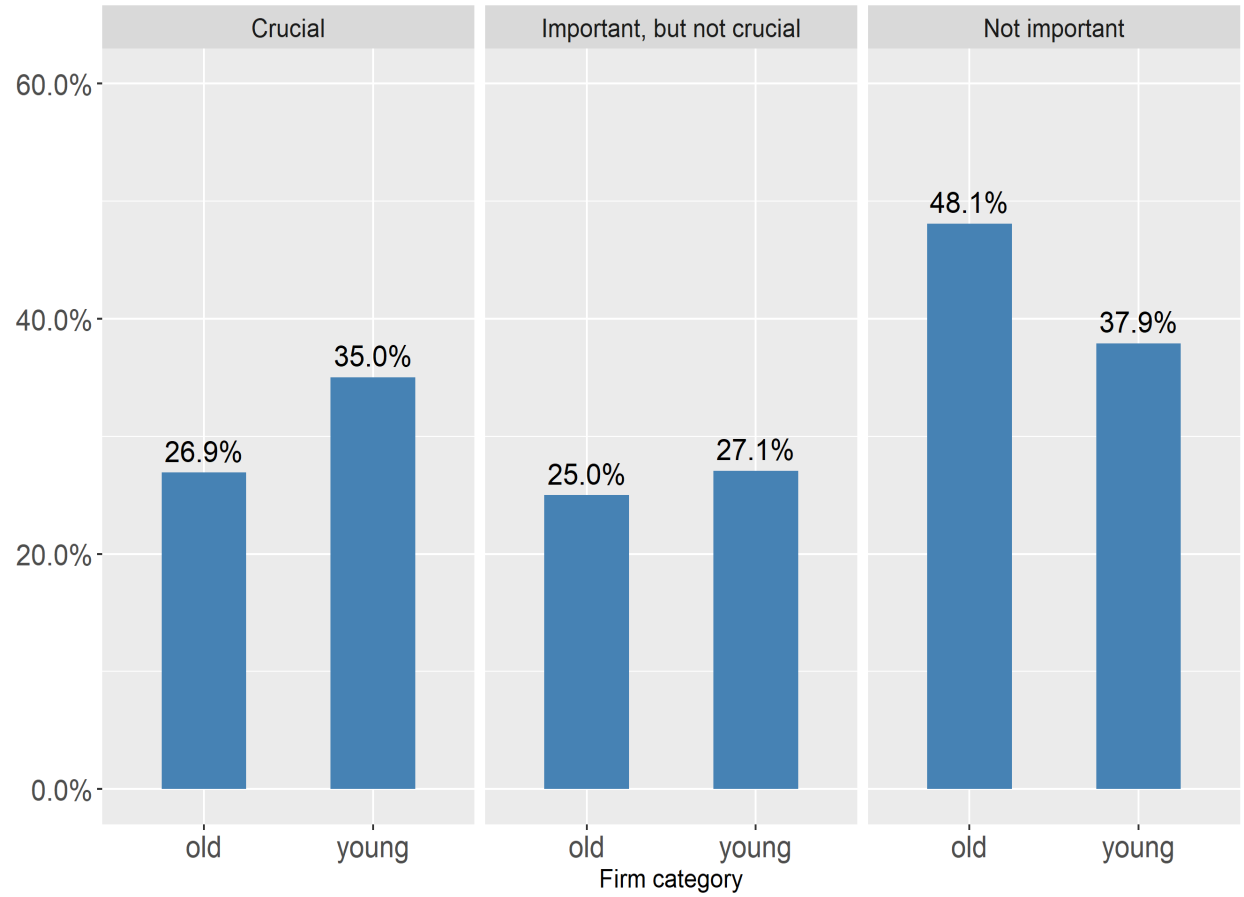
Figure 8.16 Importance of OSS for decision to start company



Note: Based on Question 14: 'How important was the existence of open-source products in your decision to start the company?' Single choice possible. Only firms that chose options a), b) or c) in Question 6. N = 1 254.

As shown in Figure 8.17, 35 % of young firms consider the availability of OSS to be crucial for their decision to start the business, whereas only 27 % of older firms attach such crucial importance of OSS for their market entry. This difference between old and young firms is statistically significant on the 95 % level. In addition, a higher share of young firms stated that the existence of OSS was an important consideration while deciding on market entry. In this case, however, the difference between young and old firms is not statistically significant.

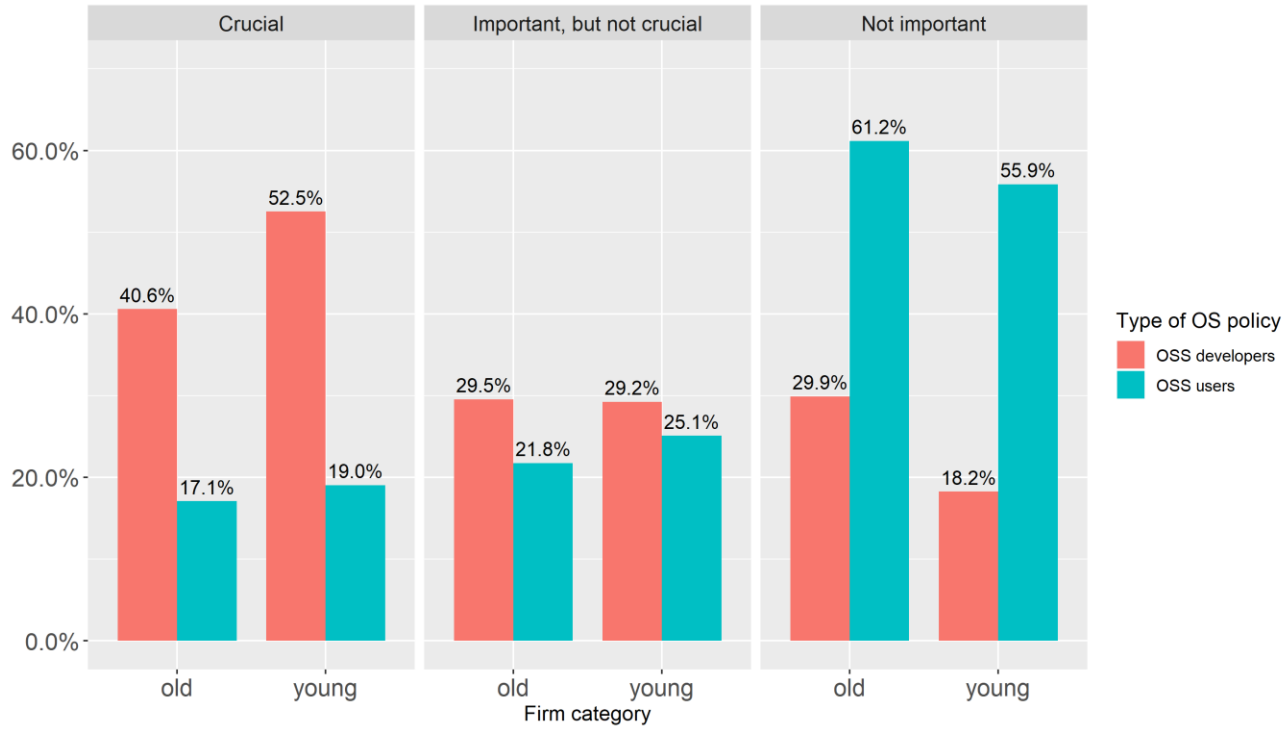
Figure 8.17 Importance of OSS for decision to start company by age of firm



Note: Based on Question 14: 'How important was the existence of open-source products in your decision to start the company?' Single choice possible. Only firms that chose options a), b) or c) in Question 6.
Young firms defined as those with their year of establishment after 2012.
N = 1 253; N young = 296; N old = 957.

However, as shown in Figure 8.18, the existence of OSS was a much more important trigger for starting the business for OSS developers than for OSS users. Whereas, over 50 % of the young and over 40 % of old firms actively developing OS software during working hours stated that OSS was crucial for that decision, the corresponding proportions among OSS users were 19 % and 17 % respectively.

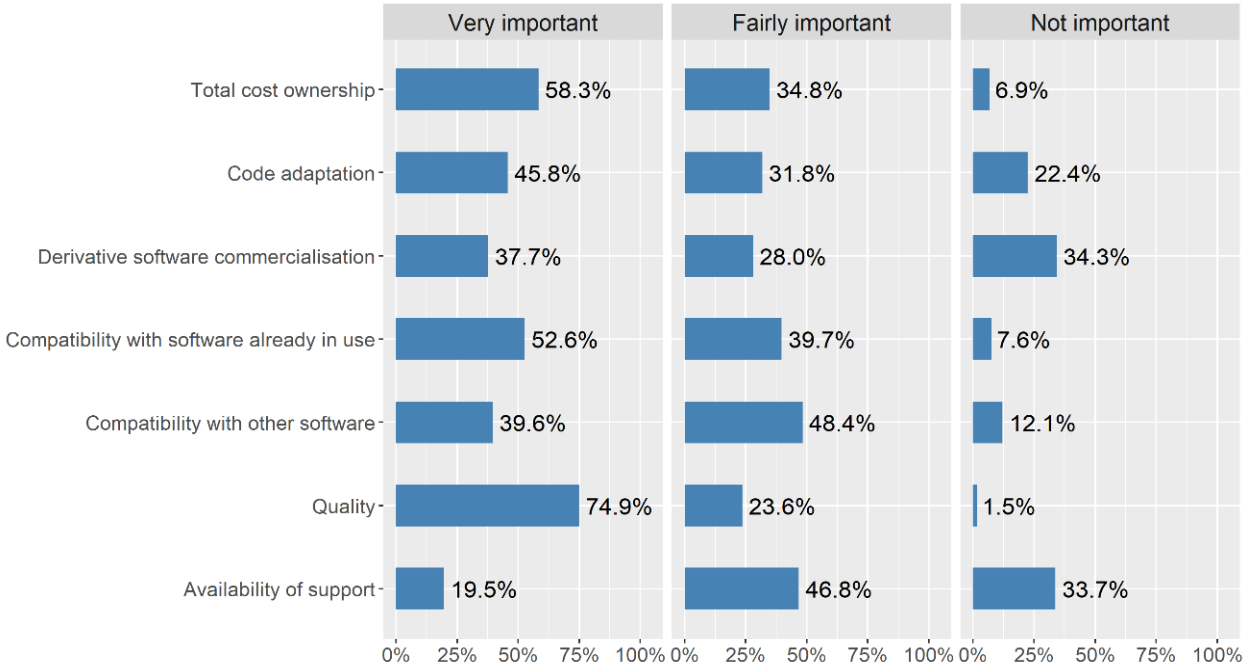
Figure 8.18 Importance of OSS for decision to start company by age of firm and OSS policy



Note: Based on Question 14: 'How important was the existence of open-source products in your decision to start the company?' Single choice possible. Only firms that chose options a), b) or c) in Question 6.
Young firms defined as those with their year of establishment after 2012.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) or c).
N = 1 253, N OSS developers, young = 145, old = 396; N OSS users, young = 151, old = 561.

The factors taken into account while making decisions on the adoption of new software in the company were also analysed. The overall results for the whole sample can be seen in Figure 8.19. Quality is the feature singled out as very or fairly important by 98.5 % of the respondents. Also high in rank of importance are total cost of ownership and compatibility with other software already in use in the firm, which more than 90 % of firms in the sample indicated as very or fairly important. Among the less important features were the possibility to freely adapt the code to the firm's needs, the availability of support and the possibility to commercialise derivative software.

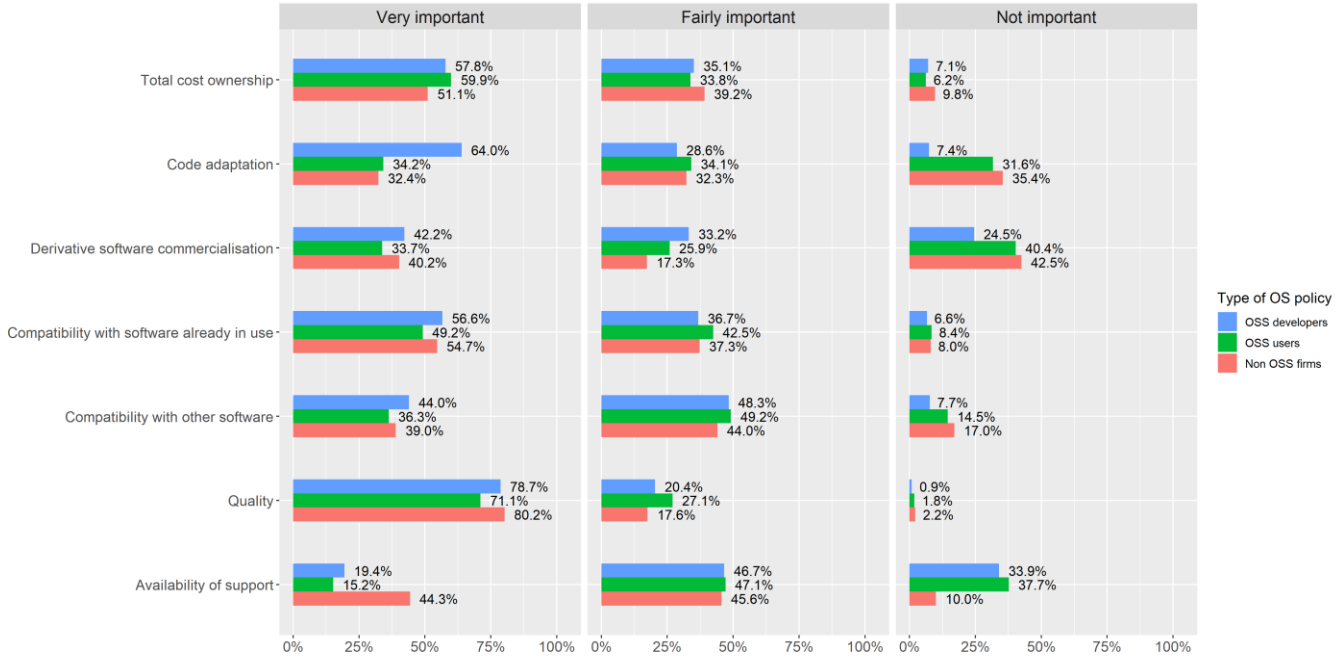
Figure 8.19 Factors taken into account when adopting new software



Note: Based on Question 16: 'What are the most important considerations when deciding to adopt new software at your company' Single choice for each statement. N = 1 364.

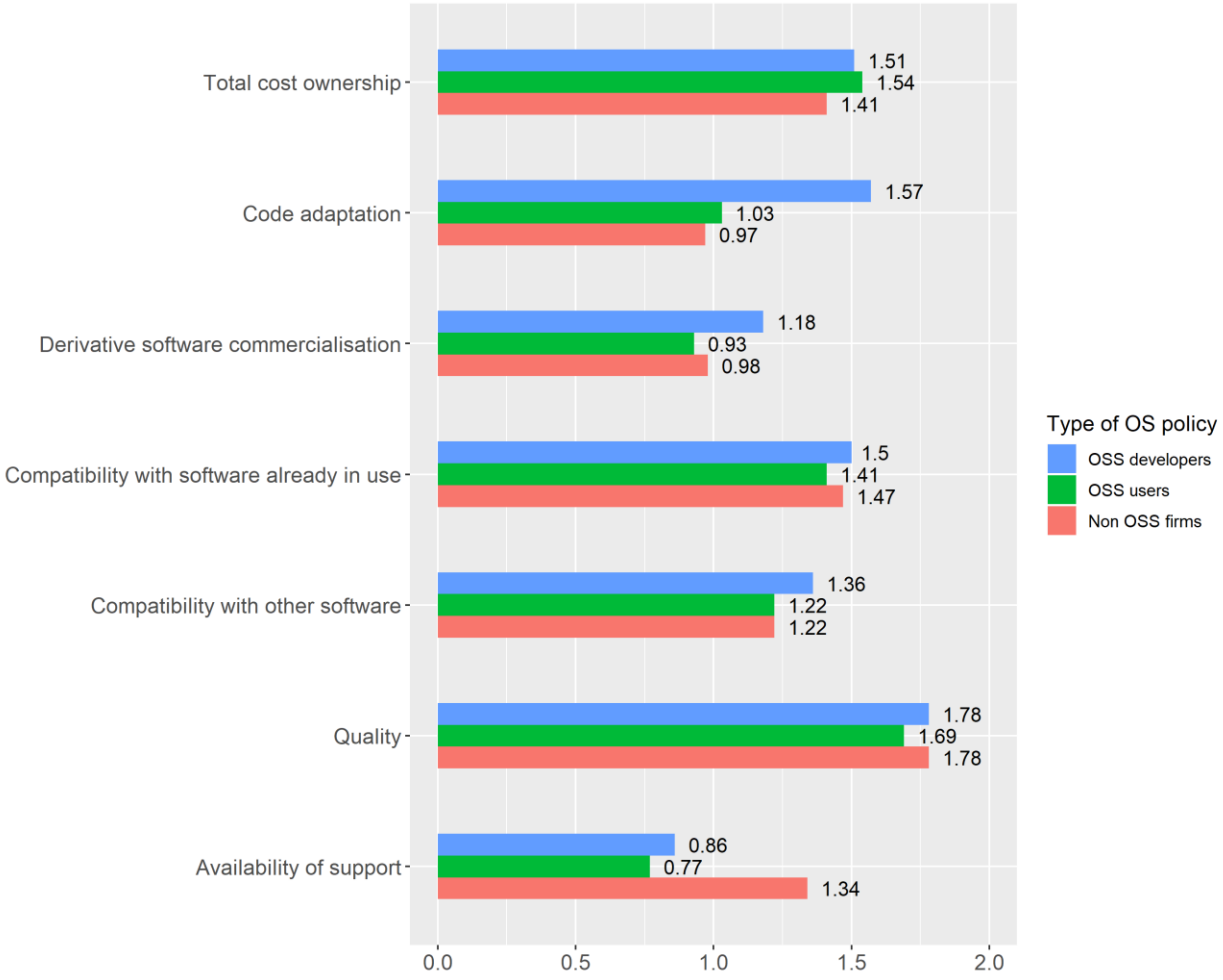
However, Figure 8.20 reveals that there are differences between OSS developers, OSS users and non-OSS firms as regards the importance of some factors, particularly those that ranked low in the general sample analysis. The possibility to freely adapt code to the firm's own needs is important to over 92 % of OSS developers (very important to 64 %), while it is not important to approximately a third of the firms categorised as OSS users and non-OSS firms. The possibility to commercialise the derivative software created by a firm is very important or fairly important for three quarters of OSS developers but for less than 60 % of OSS users and non-OSS firms. Conversely, firms that do not use or develop OS software put a much larger emphasis on the availability of support, which is important for 90 % of firms in these two groups. In contrast, the availability of support is not important for 37.7 % of OSS users and 34 % of OSS developers.

Figure 8.20 Factors taken into account when adopting new software - OS policy differences



Note: Based on Question 16: 'What are the most important considerations when deciding to adopt new software at your company?' Single choice for each statement. Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 1 364; N OSS developers = 542; N OSS users = 712; N Non-OSS firms = 110.

Figure 8.21 Quantitative analysis of factors taken into account when adopting new software - OS policy differences



Note: Based on Question 16: 'What are the most important considerations when deciding to adopt new software at your company?' Single choice for each statement. Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
Not important = 0, Fairly important = 1, Very important = 2.
N = 1 364; N OSS developers = 542; N OSS users = 712; N Non-OSS firms = 110.

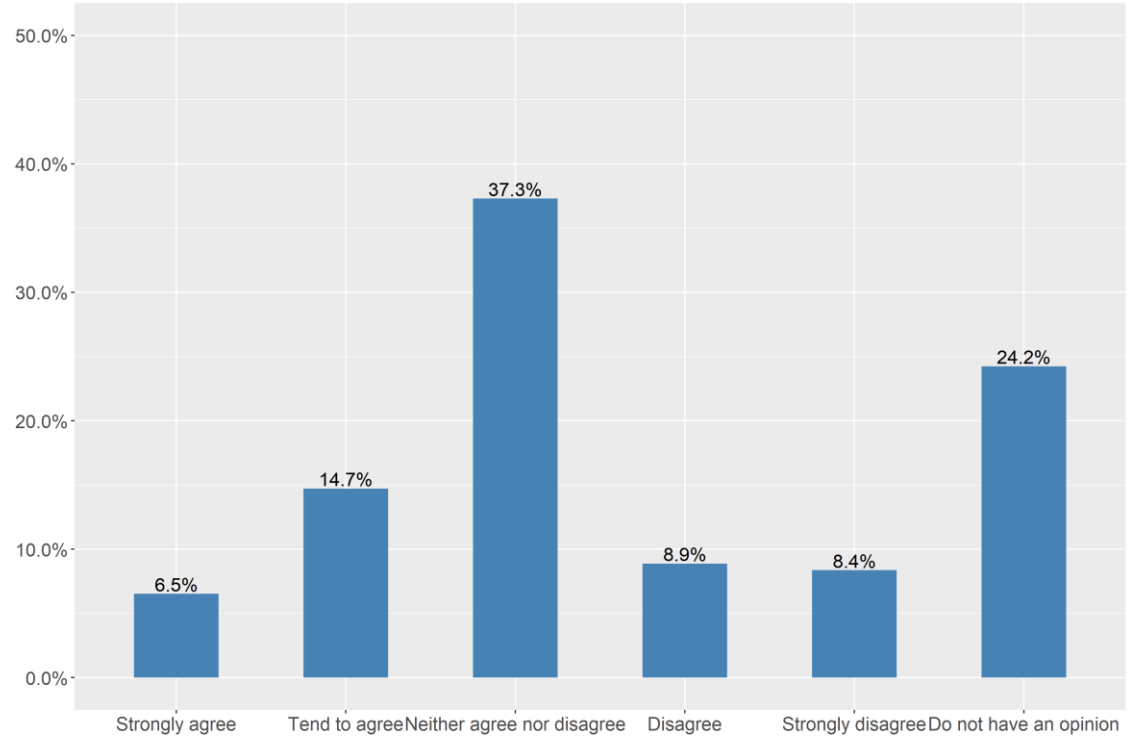
As illustrated in Figure 8.21, quality is the most important feature of the software for all firms, regardless of the type of OSS policy. However, while code adaptation is the second most important option for OSS developers, it is among the least important features for non-OSS firms. However, while availability of support is ranked high among the non-OSS firms, it is the least important option for both OSS developers and OSS users.

8.3 ROLE OF IP PROTECTION

As discussed in Section 1, firms need to be able to appropriate a sufficient proportion of the returns from their innovation in order to invest in innovative activities. The traditional view among economists is that intellectual property rights provide such incentives by allowing for temporary exclusivity in the exploitation of innovative products or services. However, there are also some informal ways of increasing appropriation. Previous research showed that informal appropriability measures may be more popular than formal ones among firms in general, not only in the software sector. This section analyses the perception of IPR protection by firms in the sample, their use of different measures of protection and their assessment of their effectiveness.

As can be seen in Figure 8.22, most respondents in the sample do not have a strong opinion on whether the current system of IPR protection is beneficial to their firm. However, among those that were able to assess it positively or negatively, positive opinions prevail. More than 21 % of respondents agreed or tended to agree with the statement that the current protection system is beneficial to their firm, whereas 17.3 % disagreed or tended to disagree with this statement.

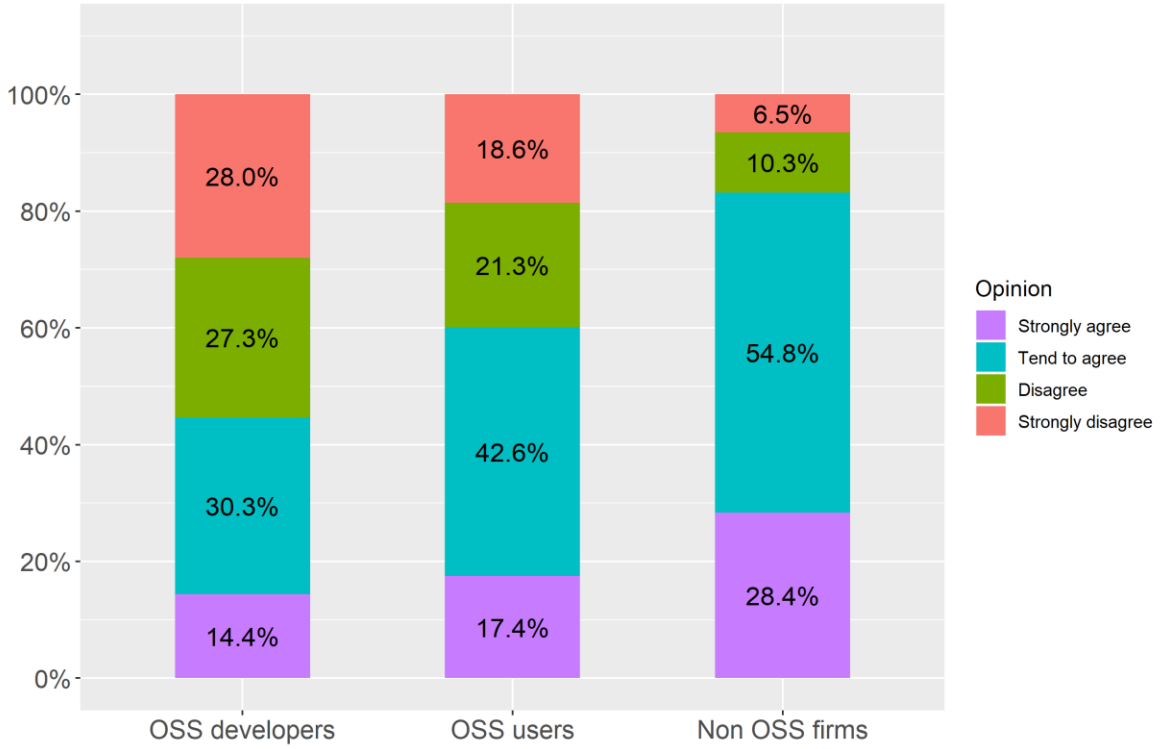
Figure 8.22 Opinions on the statement: 'the current system of IPR protection is beneficial to my firm'



Note: Based on last statement of Question 22: 'The current system of IPR protection is beneficial to my firm'. Single choice.
N = 1 321.

There is a wide difference of opinion on the current system of IPR protection depending on the firm's policy towards OSS. These differences are illustrated in Figure 6.23 which shows the answers to the same question as Figure 6.22 but only for the firms that expressed an opinion. Within this group, 44.7 % of OSS developers agreed or strongly agreed with the statement that the current system of IPR protection is beneficial to their firm. More than 60 % of OSS users and 83.2 % of non-OSS firms have a positive opinion on the benefits of the current IPR system.

Figure 8.23 Opinions on the statement: 'the current system of IPR protection is beneficial to my firm'



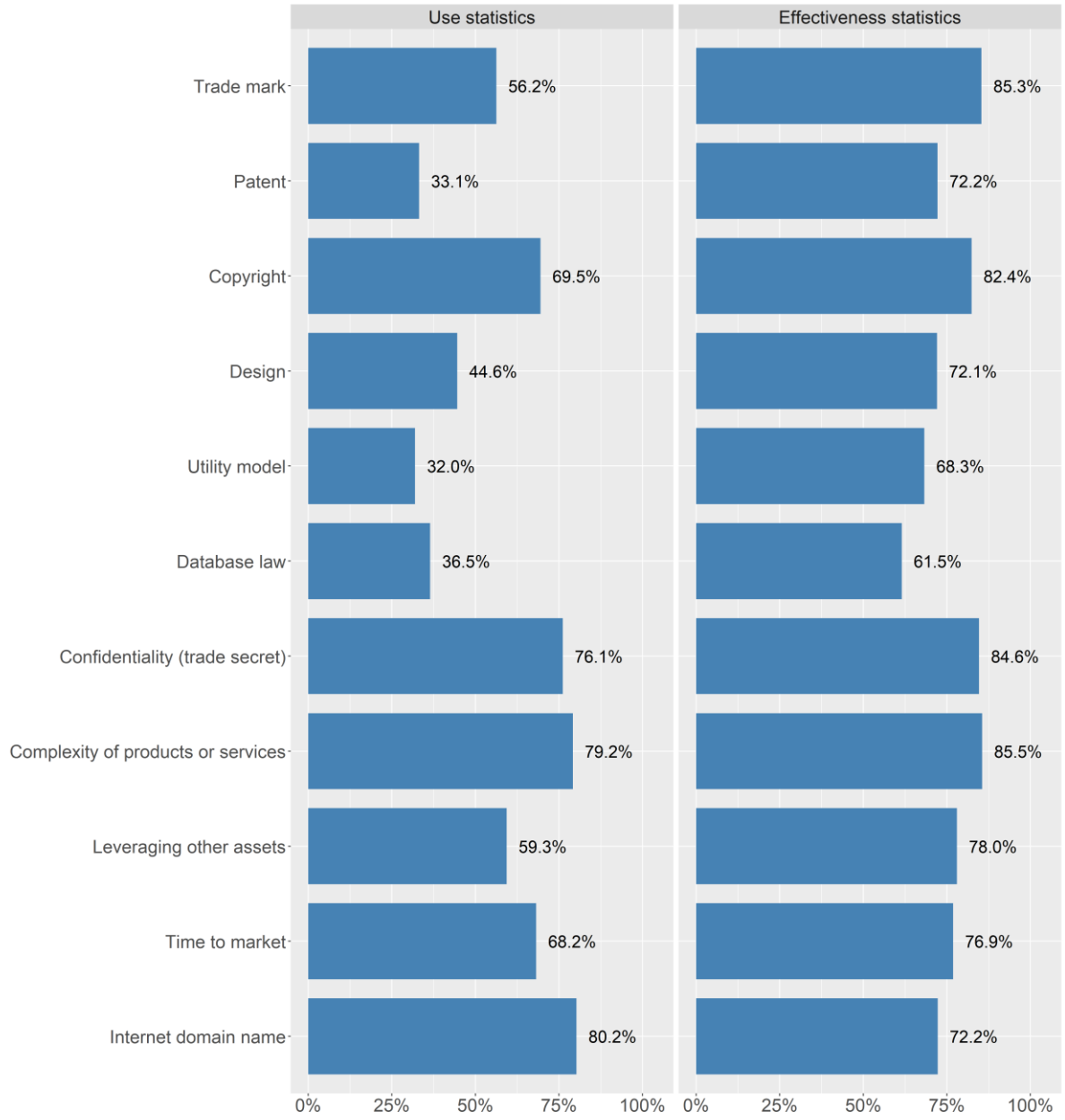
Note: Based on last statement of Question 22: 'The current system of IPR protection is beneficial to my firm' Single-choice answers 'Do not have an opinion' and 'Neither agree nor disagree' have been eliminated.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 528, N OSS developers = 225; N OSS users = 271; N Non-OSS firms = 32.

Figure 8.24 presents an analysis of the use and assessment of effectiveness of various measures of protection of products or services reported by the firms in the sample. As seen in the left-hand side panel of Figure 8.24, firms in the sample are using a combination of different formal and informal means of protection to increase the appropriation of returns from their innovation. Internet domain name is the most popular method of protection among the firms in the sample. It is used

by more than 80 % of firms. Also highly ranked, and used by more than 75 % of firms, are complexity of products or services and confidentiality. Copyright is used by close to 70 % of the firms. Next in the ranking are informal methods of protection, such as time to market (68.2 %) and leveraging other assets (59.3 %). Trade marks are used by 56.2 % of firms, registered designs by 44.6 %, database law by 36.5 %, and patents by 33.1 %. Utility models are the rights with the lowest share of users among the respondents with 32 % of firms reporting using this IPR for the protection of their products or services.

The right-hand side panel presents the firms' assessment of the effectiveness of various measures of protection. Effectiveness of all the measures of protection is assessed to be relatively high, with most of the firms using any given measure considering it to be highly or moderately effective for the protection of their products or services. The highest in the ranking of effectiveness are complexity of products or services, trade marks, confidentiality and copyright with over 80 % of firms that use those measures assessing them to be highly or moderately effective. Leveraging other assets, time to market, internet domain name, patents and designs were found to be effective measures of protection by more than 70 % of respondents. The lowest level of effectiveness by the sampled firms was assigned to utility models and database law.

Figure 8.24 Use and effectiveness of various measures of IP protection



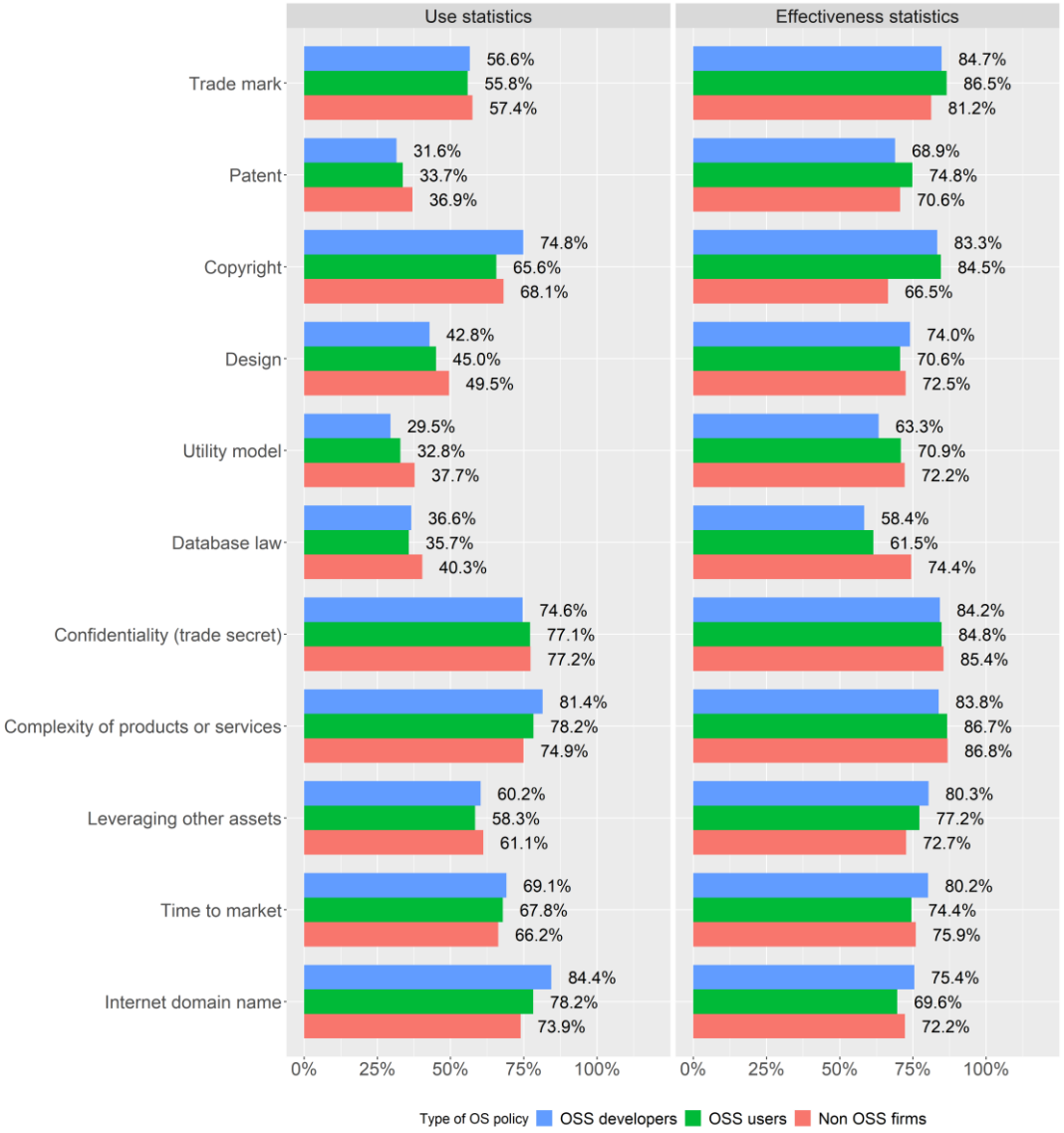
Note: Based on Question 20: 'How do you rate the effectiveness of various measures of protection for your products and/or services?' Single choice per measure of protection.
Users identified as those firms indicating the following options: 'Highly effective', 'Moderately effective', 'Not effective at all'; Non-users as firms indicating option 'I do not use it'.
Right-hand side panel shows shares of firms choosing options 'Highly effective' and 'Moderately effective' among users.
N = 1 364.

It is also interesting to investigate whether the use and assessment of effectiveness of different protection measures depends on the OSS policy of the firm. This analysis is shown in Figure 8.25. In general, there are few significant differences in the use of the various measures of protection among OSS developers, OSS users and non-OSS firms. Among non-OSS firms, the frequency of usage of formal IP rights tends to be higher than among OSS users or developers. In many cases, however, those differences are relatively low. Copyright is a notable exception, as the frequency of usage of copyright as a means of protection is 6.7 percentage points higher among OSS developers than among non-OSS firms, although this difference is not statistically significant on the 95 % level. OSS developers also tend to rely more on internet domain names as a protection measure than the other groups.

The assessment of effectiveness of various measures of protection of goods and services, shown in the right-hand side panel of Figure 8.25, is also similar between OSS developers, OSS users and non-OSS firms. Copyright is again a notable exception, as the perception of its effectiveness for protection is much higher among OSS developers and OSS users than among non-OSS firms. In this case, the difference between OSS developers and non-OSS firms amounts to 16.8 percentage points and is statistically significant on the 95 % level. The share of firms assessing trade marks and designs as effective measures of protection of their products and services is also higher among OSS-developing firms than among non-OSS firms.

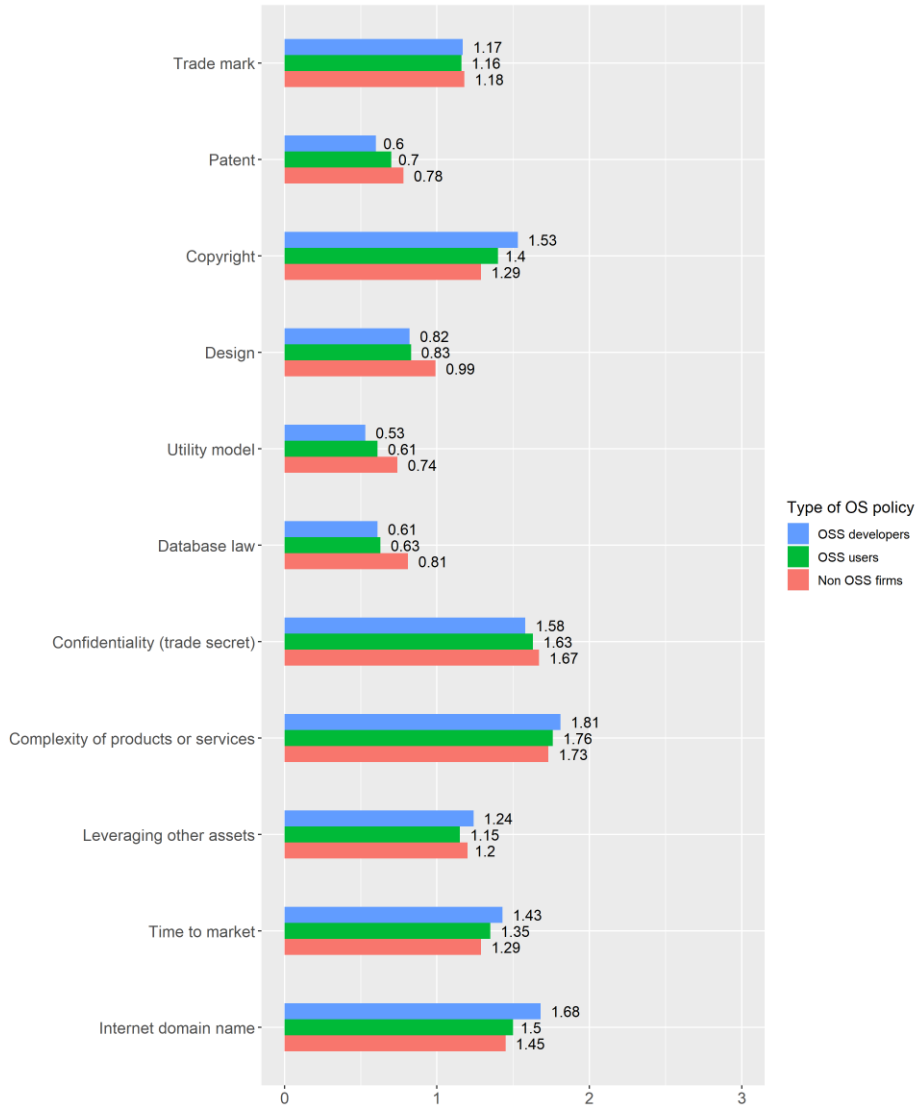
The extant research suggests that firms exhibiting a more open attitude towards their innovation may compensate the lower appropriability regime with more reliance on complementary assets. Teece (1986) named brands as one such complementary asset that can be leveraged to capture more profits from innovation. Notably, the trade mark is a measure found to be highly or moderately effective by the highest share of the OSS developers, but it is closely followed by confidentiality, complexity of products and services, copyright, leveraging other assets and time to market. For all those measures, the share of OSS-developing firms assessing them to be highly or moderately effective exceeds 80 %. Database law is the measure with the lowest perceived effectiveness among OSS-developing firms. In this case the difference between OSS developers and non-OSS firms, which use database law more frequently, amounts to 16 percentage points.

Figure 8.25 Use and effectiveness of various measures of IP protection by firms with different OSS policy



Note: Based on Question 20: 'How do you rate the effectiveness of various measures of protection for your products and/or services?' Single choice per measure of protection.
Users identified as those firms indicating the following options: 'Highly effective', 'Moderately effective', 'Not effective at all'; Non-users as firms indicating the option 'I do not use it'.
Right-hand side panel shows shares of firms choosing options 'Highly effective' and 'Moderately effective' among users.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 1 364, N OSS developers = 542; N OSS users = 712; N Non-OSS firms = 110.

Figure 8.26 Quantitative analysis of use and effectiveness of various measures of IP protection by firms with different OSS policy



Note: Based on Question 20: 'How do you rate the effectiveness of various measures of protection for your products and/or services?' Single choice per measure of protection.
Highly effective = 3, Moderately effective = 2, Not effective at all = 1, I do not use it = 0.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 1 364, N OSS developers = 542; N OSS users = 712; N Non-OSS firms = 110.

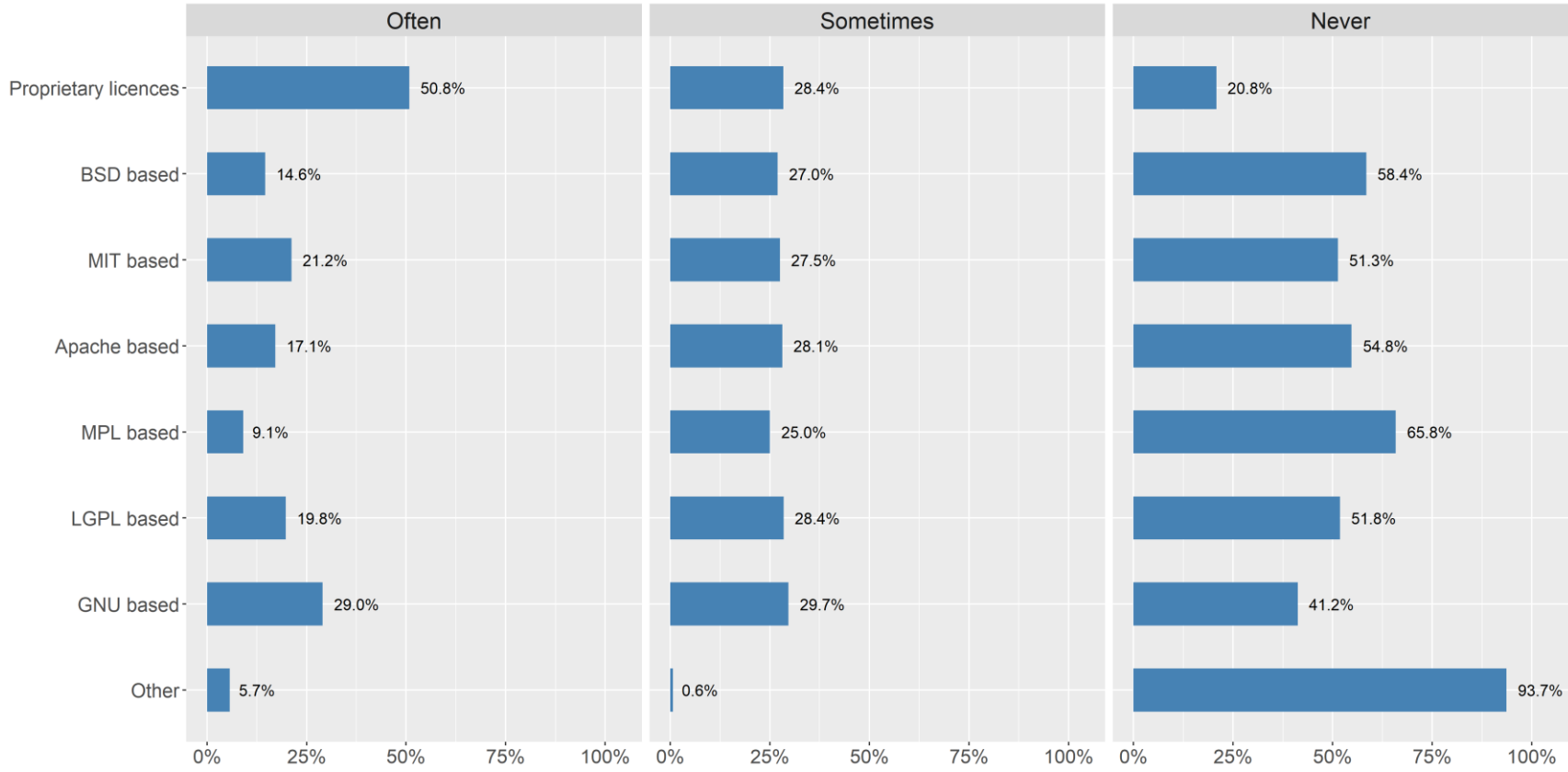
Figure 8.26 combines the use and effectiveness information into one quantitative measure. It confirms that in general the informal measures of protection are more highly rated than the formal ones by all firms, regardless of their policy towards OSS. Informal measures have higher ratings among OSS developers and users than among non-OSS firms. However, formal IP protection measures have higher ratings among non-OSS firms than among firms in the other two groups. There are, however, some exceptions to these general patterns. Trade secrets and internet domain names are rated highly by firms in all three groups. Copyright is the only IPR where the rating among the OSS developers is higher than among the non-OSS firms.

The high value OSS developers attach to copyright protection indicates that the success of the OSS model depends on a combination of copyright provisions and specific licence terms that guarantee user compliance. For many of the business models, user compliance with OSS licence terms is a crucial factor of market success as it facilitates wide dissemination and the building of a large installed base that enables sales of complementary products and services.

As shown in Figure 8.27, proprietary licences are most prevalent among the firms that distribute software developed by their employees. Surprisingly, the second most popular licence type are licences based on the GNU GPL model, which is a restrictive copyleft licence. In accordance with this licence, derivative software must be distributed under the same or equivalent terms. Other licences frequently used for the distributed software are permissive licences that do not impose strict requirements as regards the terms of redistribution licences, such as MIT, LGPL and Apache. Relatively less popular are BSD and MPL.

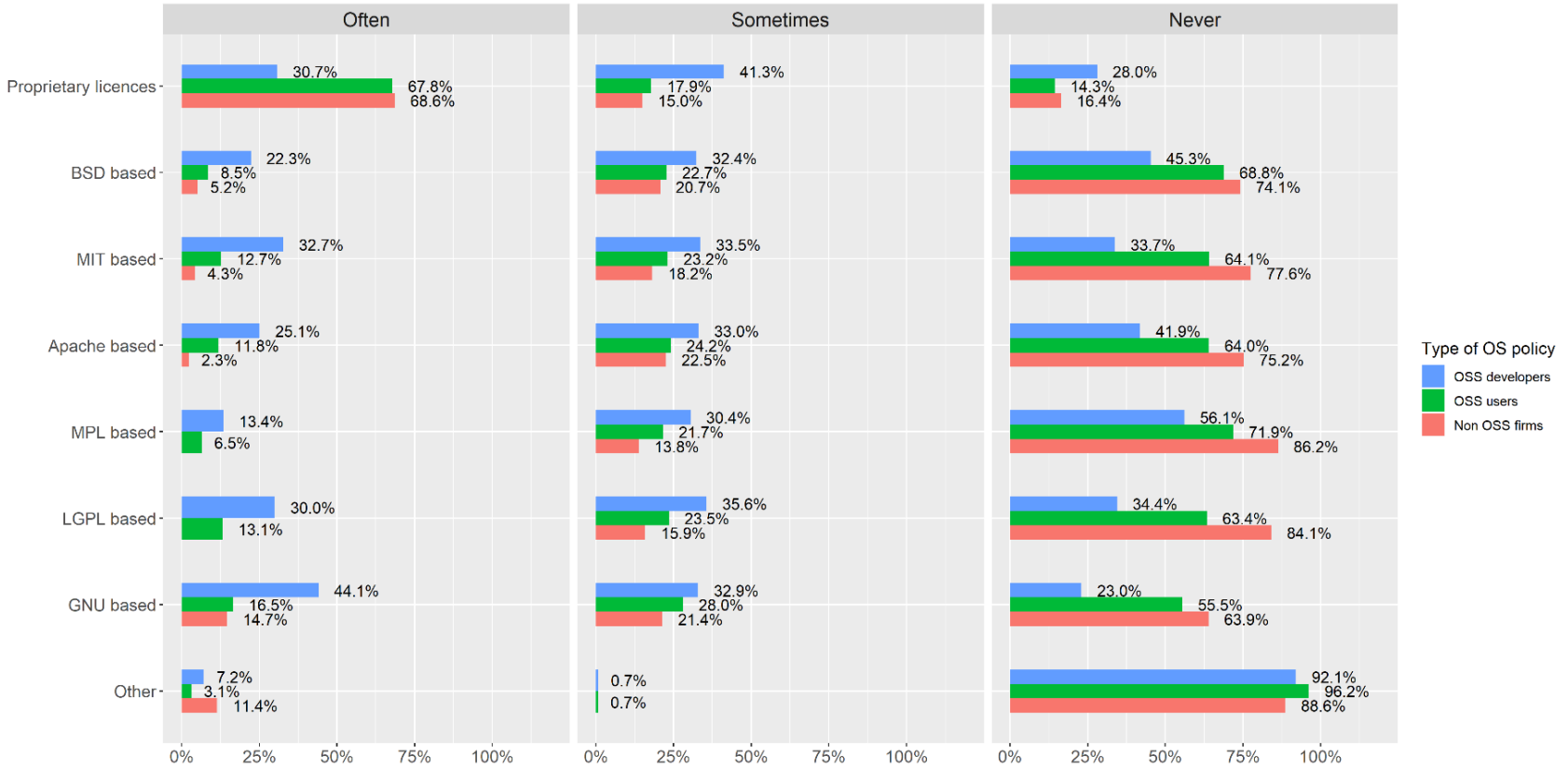
However, this general analysis conceals large heterogeneity among firms, depending on the type of policy towards OSS, shown in Figure 8.28.

Figure 8.27 Usage of licences for software distributed by firms in the sample



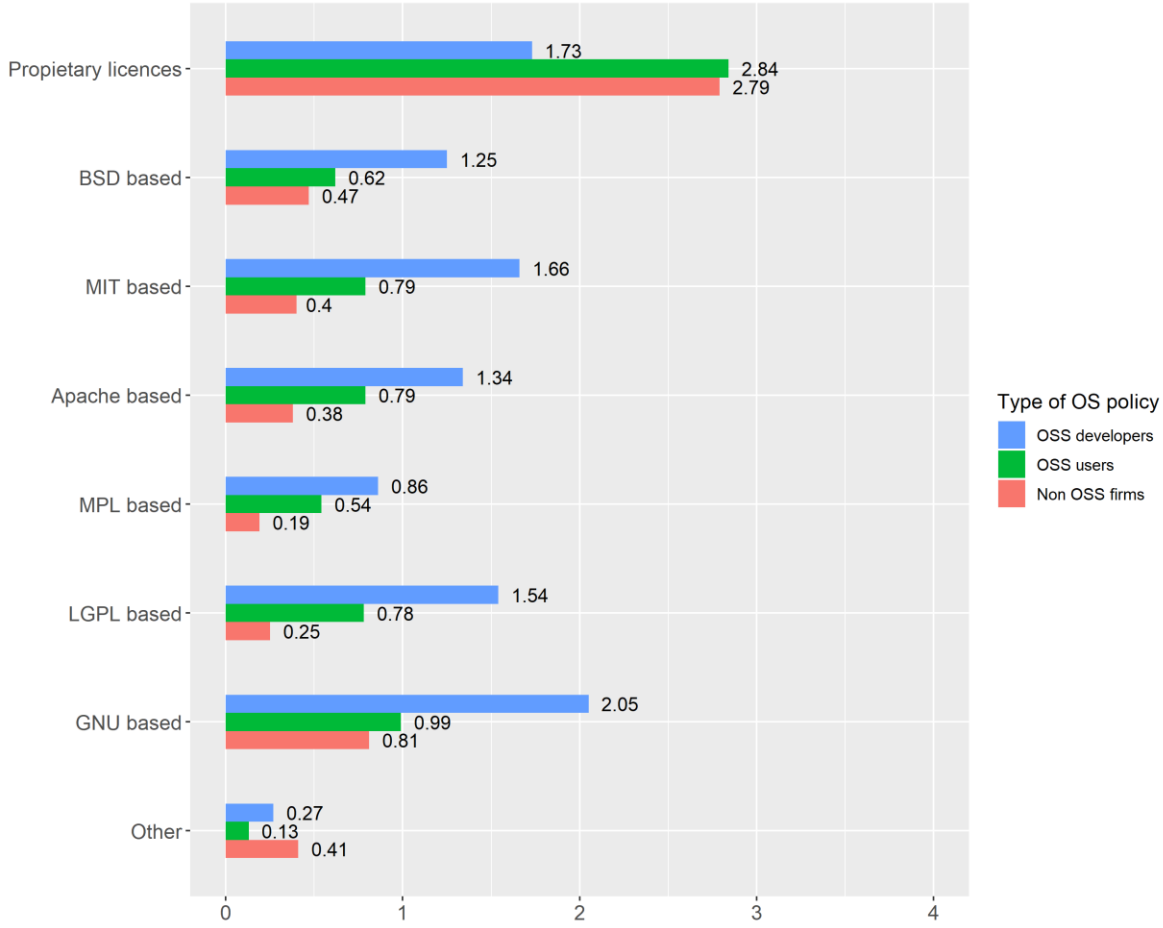
Note: Based on Question 19: If the answer to Question 17 is 'Yes'. 'What are the licence types you use to distribute the software created in your company?' Single choice for each option; Firms that chose option 'Yes' for Question 17.
Answers grouped into the following categories: 'Often' grouping respondents answering 'Very often' and 'Often', 'Sometimes' composed of respondents answering 'Sometimes' and 'Seldom' and 'Never' based on original category in the questionnaire.
N = 760.

Figure 8.28 Usage of licences for software distributed by firms in the sample - OSS policy differences



Note: Based on Question 19: 'What are the licence types you use to distribute the software created in your company?' Single choice for each option. Firms that chose option 'Yes' for Question 17.
Answers grouped into the following categories: 'Often' grouping respondents answering 'Very often' and 'Often', 'Sometimes' composed of respondents answering 'Sometimes' and 'Seldom' and 'Never' based on the original category in the questionnaire.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 760, N OSS developers = 343; N OSS users = 363; N Non-OSS firms = 54.

Figure 8.29 Quantitative assessment of the popularity of licences for software distributed by firms in the sample - OSS policy differences



Note: Based on Question 19: 'What are the licence types you use to distribute the software created in your company?' Single choice for each option. Firms that chose option 'Yes' for Question 17.
Never = 0, Seldom = 1, Sometimes = 2, Often = 3, Very often = 4.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 760, N OSS developers = 343; N OSS users = 363; N Non-OSS firms = 54.

As expected, OSS developers use proprietary licences less often than firms with more restrictive policies towards OSS. OSS developers also use more frequently various types of OSS licences than OSS users or other firms. What is surprising is that almost 15 % of firms in the category 'Non-OSS firms' claimed that they are often using GNU GPL based licences. It has to be emphasised, however, that this category consists of firms that do not allow employees to engage in development or use of OSS during working hours, as well as firms that indicated that they are not sure about their policy towards OSS and it is mainly the latter category of firms that reported using GNU GPL based licences often.

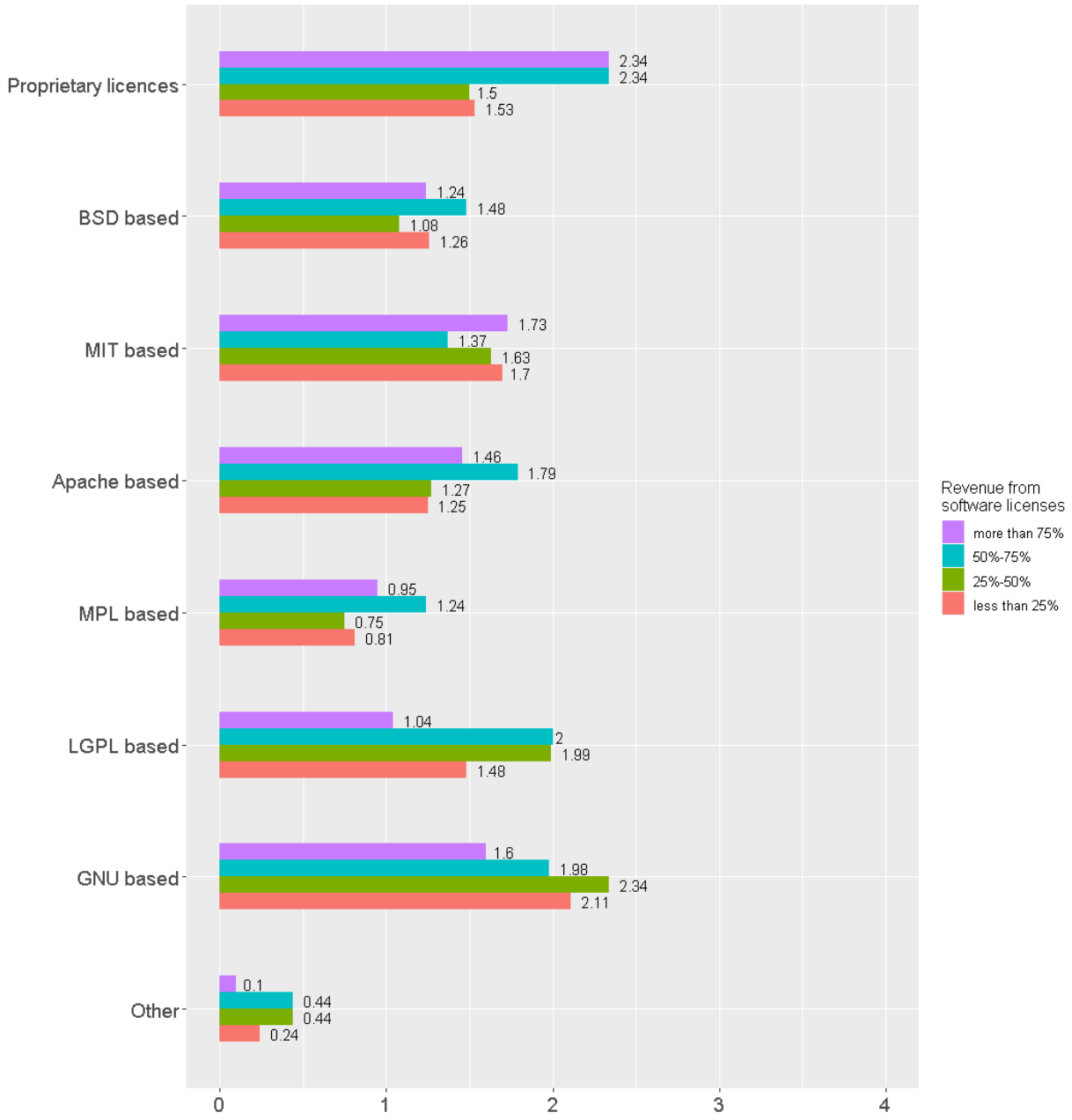
Figure 8.29 shows the quantitative assessment of various licence types by the different types of firms. Among the OSS developers, GNU based licences are used most frequently, followed by proprietary

licences and various *permissive licences:* MIT based, LGPL based, BSD based and, relatively less frequently, MPL based licences.

Although the data does not allow for an analysis of the combined usage of various licences for the same piece of software, these findings seem to confirm the conjecture of Vasudeva (2003) on the emergence of multi-licensing schemes. Under such schemes a piece of software is licensed under several licence types, combining open-source and proprietary modes. Many of those schemes bear some resemblance to the *sui generis* protection proposals formulated in the past.

Figure 8.30 shows that use of proprietary licences is not only related to the firm's OSS policy but also to the business model of the OSS-developing firms. The propensity to use proprietary licences among OSS developers differs depending on the share of revenue they derive from the sale of software licences. The higher the share of revenue from the sale of software licences, the higher the propensity to use proprietary licences among the OSS developers and the lower the propensity to distribute software under *copyleft* licences.

Figure 8.30 Quantitative assessment of the popularity of licences among OSS developers - differences depending on revenue from software licences as a share of total revenue



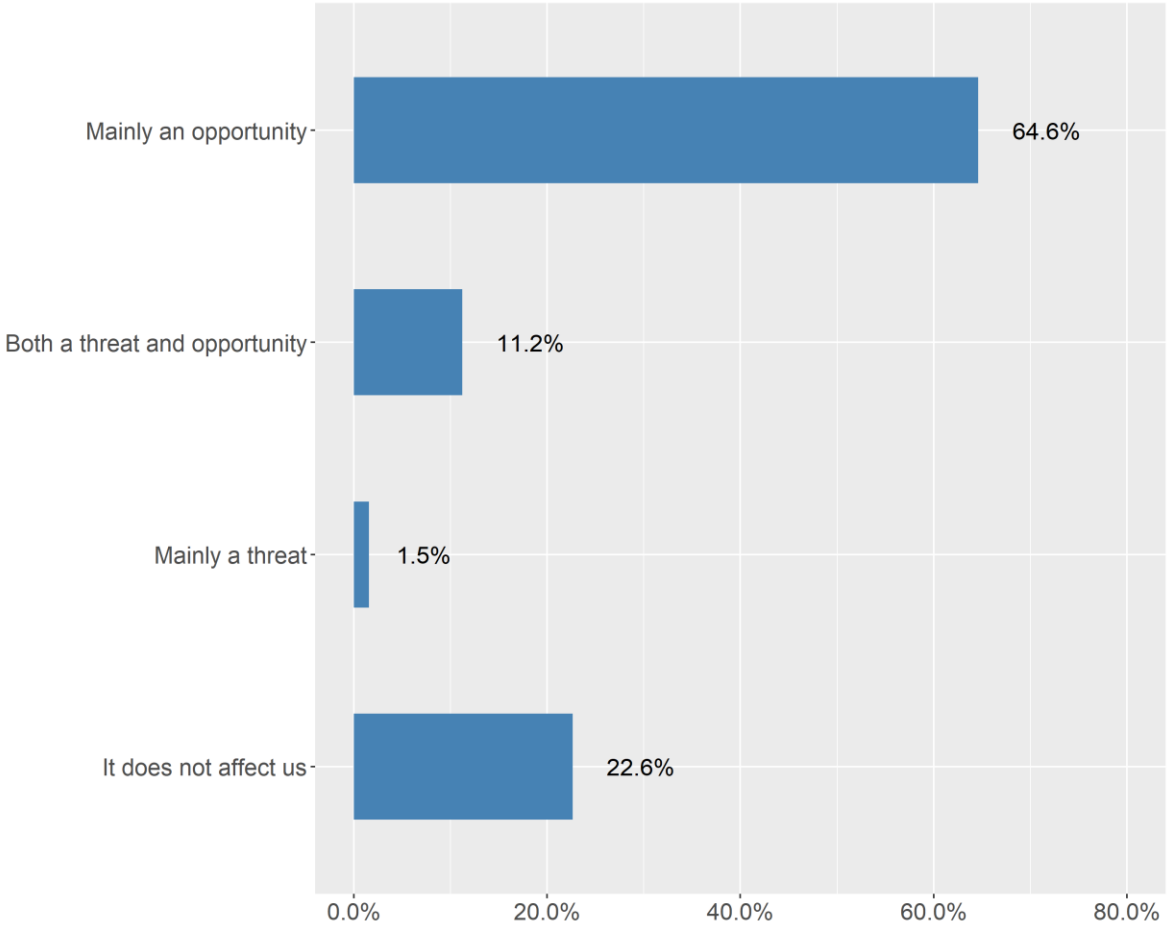
Note: Based on Question 18: If the answer to Question 17 is 'Yes'. 'What is the percentage of revenue derived from the sale of licences for software developed in your company?' Single choice. Firms that chose option 'Yes' for Questions 17; and 19: 'What are the licence types you use to distribute the software created in your company?' Single choice for each option. Firms that chose option 'Yes' for Question 17.
Never = 0, Seldom = 1, Sometimes = 2, Often = 3, Very often = 4.
Figure shows answers on OSS developers only as defined based on Question 6.
N = 343, N less than 25 % = 197; N 25-50 % = 51, N 50-75 % = 36, N more than 75 % = 59.

8.4 FIRMS' PERCEPTION OF OSS

A further set of questions in the survey explored the respondents' view of OSS in the context of their business strategy.

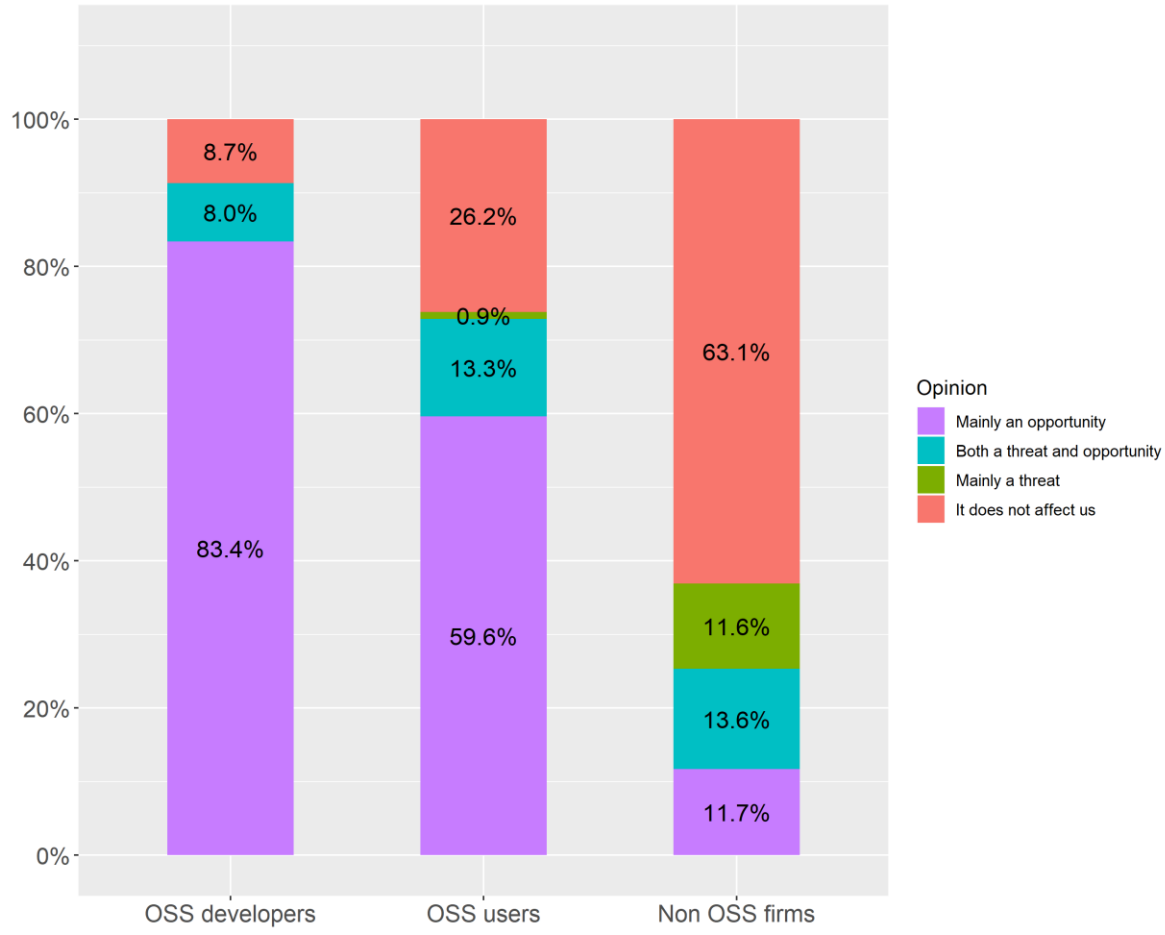
As seen in Figure 8.31, most firms in the sample see OSS mainly as an opportunity for their current business models. Only 1.5 % consider OSS to be a threat to their current business. For 22.6 % of firms OSS is not a major concern for their current business model.

Figure 8.31 Open-source products perception as regards current business model



Note: Based on Question 21: 'What is your company's perception of open-source products?' Single choice. N = 1 364.

Figure 8.32 Open-source products perception as regards current business model - variations depending on current OSS policy



Note: Based on Question 21: 'What is your company's perception of open-source products?'. Single choice. Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 1 364, N OSS developers = 542; N OSS users = 712; N Non-OSS firms = 110.

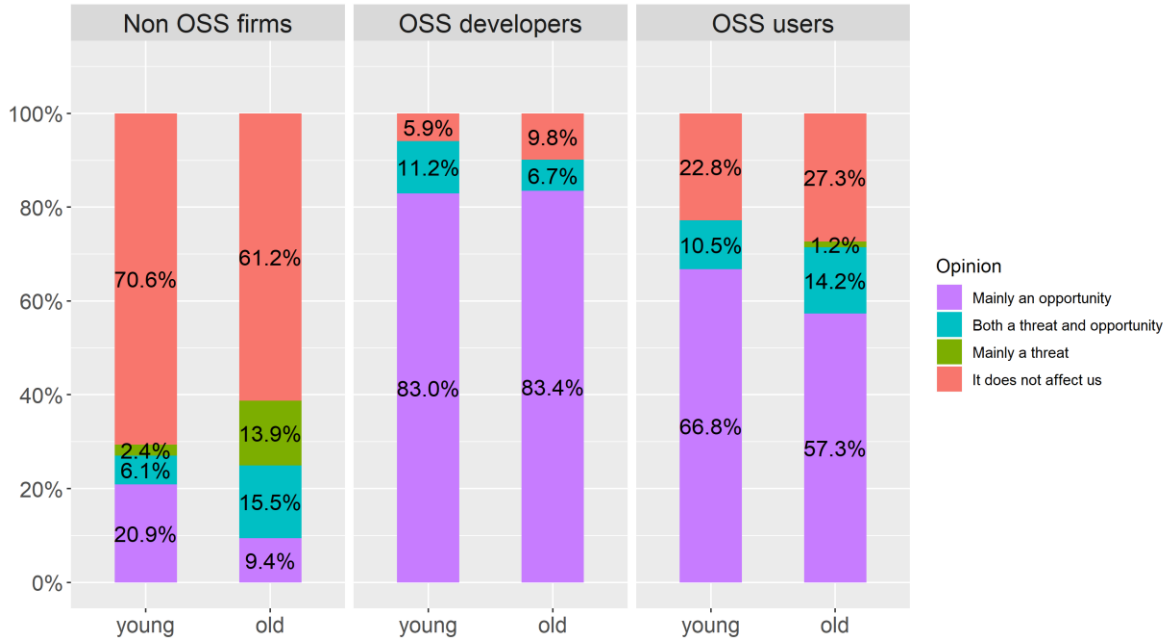
As could be expected, there are big differences in the perception of OSS as regards the current business models of firms, depending on the dominant policy of a firm towards OSS. As shown in Figure 8.32, 83.4 % of OSS developers, 59.6 % of OSS users and less than 12 % of non-OSS firms view OSS as mainly an opportunity for their current business model. However, more than 11 % of non-OSS firms sees OSS as a threat to their current business. Additionally, the share of firms that claim that OSS does not affect their business model depends on the firms' policy towards OSS. More than 63 % of firms that are not involved in OSS development or use claim that OSS does not affect their current business model. Among OSS users, the corresponding share amounts to 26.2 % and among OSS developers to 8.7 %.

However, within the group of firms that currently do not allow development or use of OS software during office hours, there are important differences in OSS perception between young and old firms. It seems that young firms are much more aware of opportunities associated with OSS. The share of firms that perceive OSS as an opportunity for their current business model is twice as high among young firms as among their

older counterparts. Conversely, the share of young firms viewing OSS as a threat to their business model is lower among young firms than among old firms in this group. Surprisingly, the share of young non-OSS firms claiming that OSS does not affect their current business model is also higher than among old firms in this group.

Among both OSS developers and OSS users not a single young firm indicated that it treats OSS as mainly a threat to its current business model.

Figure 8.33 Open-source products perception as regards current business model - variation among young and old firms and type of OSS policy

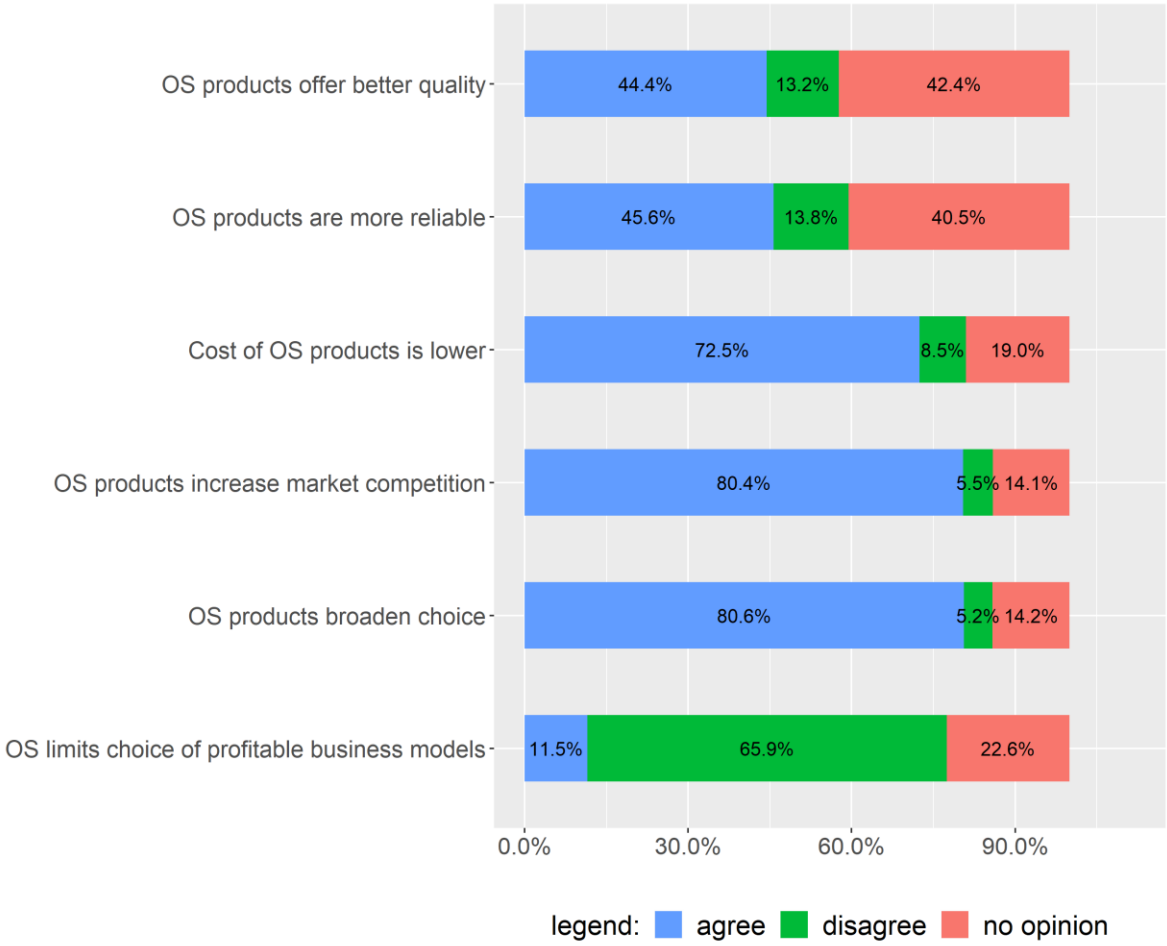


Note: Based on Question 21: 'What is your company's perception of open-source products?' Single choice.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
Young firms defined as those with their year of establishment after 2012.
N = 1 364, N OSS developers, young = 145, old = 396; N OSS users, young = 151, old = 561; N Non-OSS firms, young = 23, old = 87.

The final part of a survey consisted of a set of statements regarding the perceived attributes of OSS products. The results of this analysis are presented in Figure 8.34. Over 80 % of respondents agree with the statements that OS products increase market competition by lowering the entry barriers and enhancing the possibility of software firms to find profitable business models. Almost the same share of surveyed firms agreed that open-source significantly broadens the choice of suppliers. More than 72 % of firms agreed with the statement that the cost of OS products (including initial training and maintenance) is lower than the cost of proprietary alternatives. The statements regarding quality and reliability of OS products proved to be more controversial. 45.6 % of respondents agreed with the opinion that OS products are more reliable than proprietary alternatives and 44.4 % agreed with the statement that OS products offer better quality than proprietary alternatives. However, more than 13 % disagreed with each of those statements and more than 40 % did not have a clear opinion. Finally, 66 % of firms surveyed did not agree with the statement

that OS seriously limits the possibility of software firms to find profitable business models, with only 11.5 % of respondents sharing this view.

Figure 8.34 Perception of OSS features



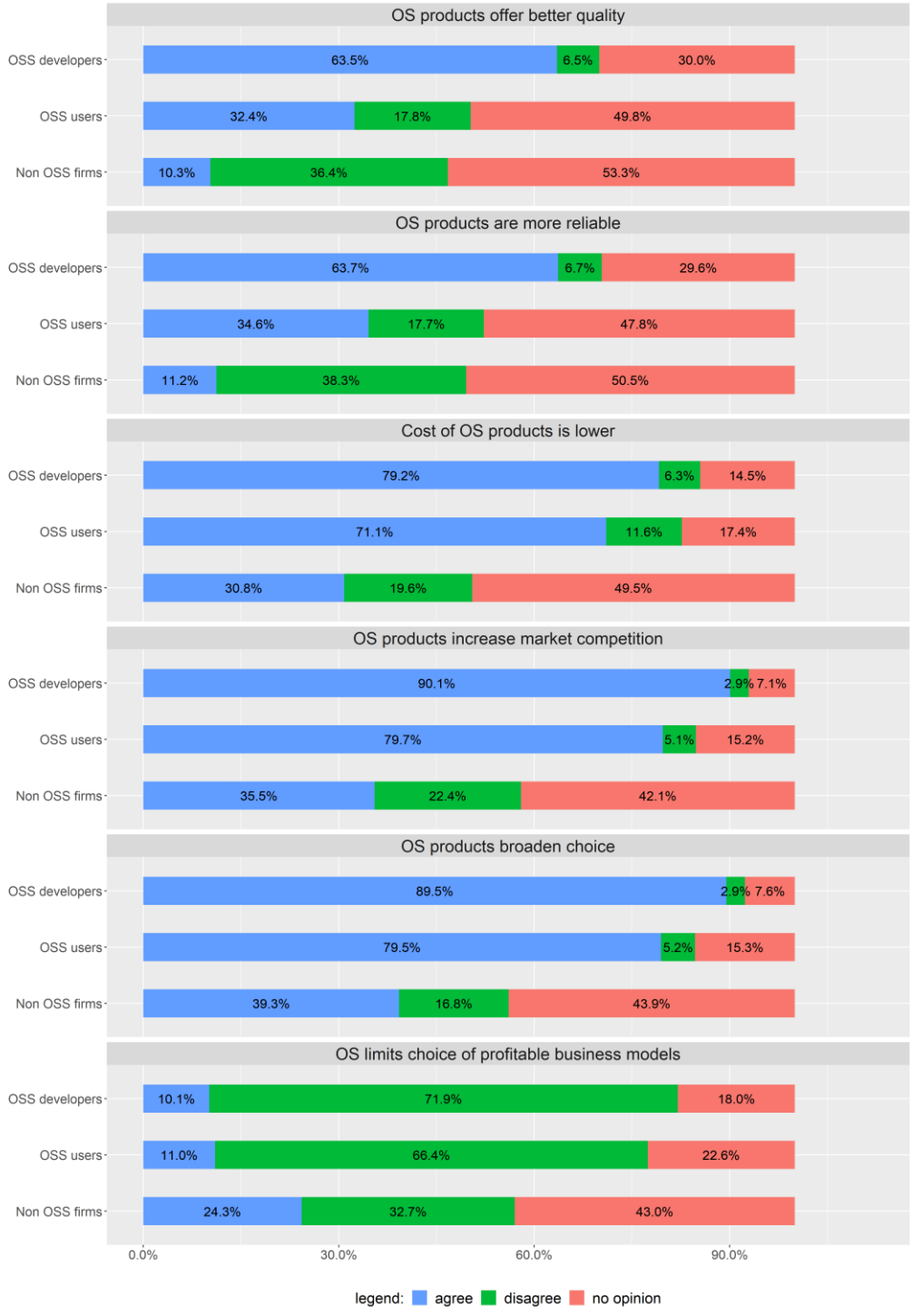
Note: Based on Question 22: 'Please provide an answer to each statement?' Single choice per statement. Answers grouped into following categories: 'Agree' grouping respondents answering 'Strongly agree' and 'Tend to agree', 'Disagree' composed of respondents answering 'Disagree' and 'Strongly disagree' and 'No opinion' based on answers 'Neither agree nor disagree' and 'Do not have an opinion'. N = 1 321.

As can be seen in

Figure 8.35, the perception of OSS is driven mainly by the OS policy within the firm. As could be expected, the share of firms with no clear opinion on various attributes of OSS is highest among firms that do not allow their employees to develop or use OSS during working hours. Such firms are more sceptical as regards OSS qualities in comparison to OSS developers or OSS users. Among non-OSS firms that do have a clear opinion, a majority disagree with the statements regarding higher reliability and better quality of OSS products. However, a majority of those firms agree about the lower cost of OS products. While they tend to agree that OS products increase market competition and broaden choice, most firms in this group disagree with the view that OS limits their choice of profitable business models.

In general, OSS users, and even more so OSS developers, are much more positive regarding qualities of OSS products and are less sceptical regarding the possibility to find profitable business models than their counterparts not developing or using OSS during working hours.

Figure 8.35 Perception of OSS features depending on the OSS policy in the firm



Note: Based on Question 22: 'Please provide an answer to each statement?' Single choice per statement. Answers grouped into following categories: 'Agree' grouping respondents answering 'Strongly agree' and 'Tend to agree', 'Disagree' composed of respondents answering 'Disagree' and 'Strongly disagree' and 'No opinion' based on answers 'Neither agree nor disagree' and 'Do not have an opinion'.
Type of OS policy defined based on Question 6: OSS developers' firms that chose option a), OSS users' firms that chose options b) and c); Non-OSS firms – other firms choosing options d) and e).
N = 1 321; N OSS developers = 523; N OSS users = 691; N Non-OSS firms = 107.

8.5 PROFILE OF FIRMS BENEFITING FROM OSS

As discussed in Section 8.1, firms that encourage their employees to develop and use OS software (OSS developers) constitute almost 40 % of the population of firms active in the software sector, as defined in the present report. In this section, the findings from the previous sections are brought together to paint a profile of a 'typical' firm that incorporates OSS into its business.

OSS developers tend to be younger than the average software firm. In most firms, employees spend less than 25 % of their working time on OSS projects. The engagement in OSS projects in those firms has remained the same or increased rather than decreased in the last three years. The share of firms distributing software among this group of firms is higher than the average in the software industry. However, for most OSS developers income from the sale of licences is less than 25 % of their total revenue.

Over 70 % of those firms use proprietary licences for some of their products, but they also tend to use a variety of licences with varying degrees of strictness as regards the terms of licensing derivative software. The most popular among them are strict copyleft GNU based licences and permissive MIT based and LGPL based licences. There is, however, a clear tendency to rely more on proprietary licences and less on copyleft licences with a higher importance of sale of software licences for the total revenue of OSS-developing firms. The higher the reliance on software licences sale in the total revenue of a firm, the higher the propensity to use proprietary licences.

OSS-developing firms are able to make a profit by embedding OSS into a variety of business models, most often with services complementing OS products, and by embedding OS software into tangible products. The existence of the OSS model was an important consideration for those firms to start business activity. Most young firms (less than 5 years' old) and more than 40 % of older OSS-developing firms stated that OSS was of crucial importance for their decision to start a business.

More than 80 % of OSS developers perceive OSS as mainly an opportunity for their business model. While they express some scepticism in their assessment of the benefits of the current IPR system for their businesses, their pattern of use and effectiveness perception of various modes of IP protection do not differ dramatically from firms with more restrictive policies towards OSS. They tend to use and perceive informal forms of protection as more effective, but differences with other types of firms, with a few exceptions, are not significant.

Interestingly, OSS developers tend to rely more on copyright as a form of protection of their IP than do firms that do not engage in OSS development. This serves as an indication that the OSS model is sustained by the creative application of copyright provisions in the OSS licences.

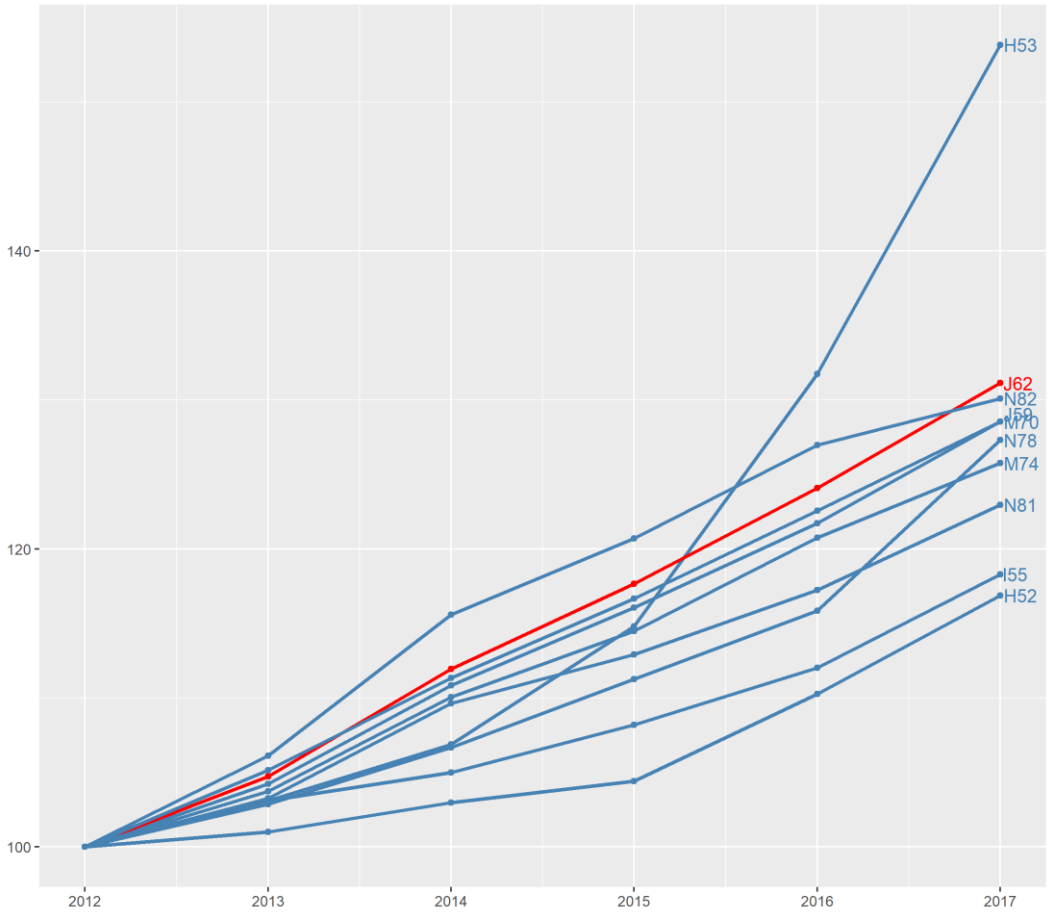
9. CONCLUSIONS

OSS has become an important and integral part of the software sector. Almost 40 % of firms surveyed stated that their employees are developing OSS products during working hours. A further 50 % of respondents encourage or allow employees to use OSS products for their work. Active participation in OSS product development allows firms to develop customised solutions for their clients more efficiently and with lower costs by building upon the creative efforts of others. Active participation in OSS communities helps firms to increase competences of their staff and thus enhance their capacity for innovation.

Development of OSS is a reaction to several factors specific to the software industry and the way IPR protection developed in this sector. To some extent it is a testimony to the resilience and versatility of the current IPR system, which allows for the emergence of new institutions and innovation governance methods. IPR provisions, especially copyright, have been used creatively by developers of OSS licences to offer free access to source code while still preserving their ability to appropriate the returns on innovation. The OSS model has not limited the possibilities to develop profitable business models in the industry, but rather it has to some extent helped shift the emphasis from software as such to products and services complementing it. Eurostat data indicates that the *Computer programming and consultancy and related activities* is one of the industries with the highest growth rates in the number of firms in recent years (see Figure 7.1 below). In addition, the mean annual 'birth rate' in this industry exceeded 13 % over the period of 2012 to 2017[(26)]. This data indicates that despite, or maybe thanks to the existence of OSS, there are still plenty of opportunities that are being explored and exploited by entrants. However, as shown in the present report, for the conversion of these opportunities into profitable business models, firms engaged in the development of OSS rely on many informal and formal IP protection methods.

[(26)] Based on indicator V97020 birth rate: number of enterprise births in the reference period (t) divided by the number of enterprises active in year t (percentage) available in table bd_9bd_sz_cl_r2 'Business demography by size class (from 2004 onwards, NACE Rev. 2)'.

Figure 9.1 Top 10 NACE divisions with the highest growth of number of firms between 2012 and 2017



Legend: H53: Postal and courier activities
J62: Computer programming, consultancy and related activities
N82: Office administrative, office support and other business support activities
J59: Motion picture, video and television programme production, sound recording and music publishing activities
M70: Activities of head offices; management consultancy activities
N78: Employment activities
M74: Other professional, scientific and technical activities
N81: Services to buildings and landscape activities
I55: Accommodation
H52: Warehousing and support activities for transportation

The OSS-developing firms renounce the right for temporary exclusivity to appropriate their contribution in software, by putting their works in the common pool and making their software open for others to reuse. They are, however, able to profit from their OSS involvement by combining it with other, proprietary, services and products, complementing OSS. Proper IP protection strategies for those complementary goods and services are as important for firms engaged in OSS development as for other firms.

Although there are some differences between OSS-developing firms and other firms as regards patterns of IPR use, those differences are rather small, which suggests that all software firms rely on similar mechanisms for appropriating the returns from their creative activity.

From the survey carried out in this study, it appears that rather than the ideological motives sometimes ascribed to the OSS movement, engagement in OSS is driven by pragmatism related to lower cost and strategic advantages stemming from the decentralised process of OSS development. The firms' contribution to the OSS pool of knowledge is linked with their ability to sell the products and services complementing their contribution.

OSS governance and the effectiveness of business models based on OSS rely heavily on the users' compliance with OSS licence terms and copyright provisions. There are certain rules associated with usage of each product developed under OSS, that have to be complied with and enforced in order for this model to work properly. Therefore, there is a need for higher awareness of the functioning of OSS among policy makers, and also among law enforcement and the judiciary.

As discussed in this report, OSS emerged as a reaction of private agents to the direction of evolution of the software industry, the specificities of innovation in this sector and uncertainties regarding IPR protection of software. Therefore, it is not clear whether a similar system of knowledge production and dissemination could be generalised and applied successfully at the same scale in other sectors. This does not preclude the emergence of similar structures of innovation governance in specific sectors, where the innovation has incremental character. Similar solutions, involving putting a large part of knowledge with commercial potential into a common pool, have emerged in the biotechnology industry, where databases are disseminated to preempt appropriation and knowledge privatisation (Merges, 2004). There are also more recent examples of the open-source culture spreading to hardware such as microchips[27]. Those sectors may be interesting subjects of further research that may yield more general insights into firms' motivations behind giving up exclusivity in exploiting results of innovation and their ability to combine such open innovation with profitable commercial activity.

[27] *Your own RISC*, the Economist, 3 October 2019.

10. BIBLIOGRAPHY

Aigrain, P. (2010). An Uncertain Victory: The 2005 Rejection of Software Patents by the European Parliament, in Krikorian, G., & Kapczynski, A. ed. Access to knowledge in the age of intellectual property.

Anderson, C. (2013). Free: How today's smartest businesses profit by giving something for nothing. Random House.

Bonaccorsi, A., Giannangeli, S., & Rossi, C. (2006). Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management science*, *52*(7), 1085-1098.

Brynjolfsson, E., & Kemerer, C. F. (1996). Network externalities in microcomputer software: An econometric analysis of the spreadsheet market. Management science, 42(12), 1627-1647.

Carr, N. G. (2003). IT doesn't matter. *Harvard business review*, *81*(5), 41.

Chandler, W. (2015). Patentability of computer-implemented inventions (CIIs): state of play and developments, Supplementary publication – Official Journal EPO, 5/2015.

Comino, S., Manenti, F. M., & Thumm, N. (2015). Intellectual property and innovation in information and communication technology (ICT). *European Commission, JRC Science for Policy Report no. JRC97541*.

Comino, S., Manenti, F. M., & Thumm, N. (2019). The role of patents in information and communication technologies: a survey of the literature. *Journal of Economic Surveys*, *33*(2), 404-430.

Crowcroft J. *Copyright, piracy and software* in Bently, L., Davis, J., & Ginsburg, J. C. (Eds.). (2010). *Copyright and Piracy: An interdisciplinary critique* (Vol. 13). Cambridge University Press.

Davies, S. (2003). The Proposed Software Directive: A User's Comments, 2003 (1) The Journal of Information, Law and Technology (JILT).

Diaz, Gerardo Con. Software Rights: How Patent Law Transformed Software Development in America. Yale University Press, 2019.

Foray, D. (2006). *Economics of knowledge*. MIT press.

Fox, R. (2013). *Information Technology: An Introduction for Today's Digital World*. CRC Press.

Frietsch, R., Neuhäusler, P., Melullis, K. J., Rothengatter, O., & Conchi, S. (2015). The economic impacts of computer-implemented inventions at the European Patent Office. *Fraunhofer Institute for Systems and Innovation Research*.

Ghosh, R. A. (2005). Understanding free software developers: Findings from the FLOSS study. *Perspectives on free and open source software*, 23-46.

Gomulkiewicz, R. W. (2018). Is the License Still the Product. *Ariz. L. Rev.*, *60*, 425.

Graham, S. J., & Mowery, D. C. (2003). Intellectual property protection in the US software industry. *Patents in the Knowledge-based Economy*, *219*, 231.

Graham, S. J., Merges, R. P., Samuelson, P., & Sichelman, T. (2009). High technology entrepreneurs and the patent system: Results of the 2008 Berkeley patent survey. *Berkeley Technology Law Journal*, *24*(4), 1255.

Gray-Le Coz, T., & Duan, C. (2014). Apply it to the USPTO: Review of the Implementation of Alice v. CLS Bank in Patent Examination. *Patently-O Patent LJ*, *2014*, 1-3.

Haff, G. (2018). How Open Source Ate Software. Understand the Open-Source Movement and So Much More. Apress.

Haunss, S. (2013). *Conflicts in the knowledge society: the contentious politics of intellectual property* (Vol. 20). Cambridge University Press.

Hall, B., Ziedonis, R. The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semi-Conductor Industry, 1979-1995. *The RAND Journal of Economics*, February 2001.

Harison, Elad. (2008). Intellectual property rights, innovation and software technologies: the economics of monopoly rights and knowledge disclosure. Edward Elgar Publishing.

Hecker, F. (1999). Setting up shop: The business of open-source software. *IEEE software*, *16*(1), 45-51.

Huang, K. G. L. (2016). Uncertain intellectual property conditions and knowledge appropriation strategies: Evidence from the genomics industry. *Industrial and corporate change*, 2017, Vol. 26, No. 1, 41-71.

Husovec, M. (2018) Standardization, Open Source, and Innovation: Sketching the Effect of IPR Policies, TILEC Discussion Paper, DP 2018-034.

Katz, M. L., & Shapiro, C. (1985). Network externalities, competition, and compatibility. *The American economic review*, *75*(3), 424-440.

Koukal, P. (2018). GRAPHICAL USER INTERFACES AND THEIR PROTECTION IN THE EUROPEAN UNION. *Current Politics & Economics of Europe*, *29*(4).

Kretschmer, M. (2003). Software as text and machine: the legal capture of digital innovation. *Journal of Information, Law and Technology*, *1*.

Lakhani, K. & Wolf, R. (2005). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. in Joseph Feller, Brian Fitzgerald, Scott A. Hissam, Karim R. Lakhani (Eds.) *Perspectives on Free and Open Source Software,* Cambridge, MA: The MIT Press.

Lerner, J., & Tirole, J. (2002). Some simple economics of open-source. *The journal of industrial economics*, *50*(2), 197-234.

Lerner, J., & Tirole, J. (2005). The economics of technology sharing: Open-source and beyond. *Journal of Economic Perspectives*, *19*(2), 99-120.

Li, Y. (2019). The Current Dilemma and Future of Software Patenting. *IIC-International Review of Intellectual Property and Competition Law*, *50*(7), 823-859.

Lippoldt D., Stryszowski P. (2009). *Innovation in the software sector*, Organisation for Economic Co-operation and Development.

Matusow J. (2005) Shared Source: The Microsoft Perspective in Joseph Feller, Brian Fitzgerald, Scott A. Hissam, Karim R. Lakhani (Eds.) *Perspectives on Free and Open Source Software,* Cambridge, MA: The MIT Press.

Martinez Catalina (2018), Expanding Patents in the Digital World: The Example of Patents in Software in In X. Seuba, C. Geiger, & J. Pénin (Eds.), Intellectual Property and Digital Trade in the Age of Artificial Intelligence and Big Data. Centre d'études internationales de la propriété intellectuelles (CEIPI). CEIPI / ICTSD Publication Series on Global Perspectives and Challenges for the Intellectual Property System, No. 5

Maurer, S. M., & Scotchmer, S. (2006). *Open-source software: the new intellectual property paradigm* (No. w12148). National Bureau of Economic Research.

Ménière, Y., & Thumm, N. (2015). Fair, Reasonable and Non-Discriminatory (FRAND) Licensing Terms. Research Analysis of a Controversial Concept. JRC Science and Policy Report

Merges, R. P. (2004). A new dynamism in the public domain. *The University of Chicago Law Review*, 183-203.

Morgan, L., & Finnegan, P. (2014). Beyond free software: An exploration of the business value of strategic open source. *The Journal of Strategic Information Systems*, *23*(3), 226-238.

National Research Council. (1991). *Intellectual Property Issues in Software*. National Academies Press.

Nuvolari, A. (2004). Collective invention during the British Industrial Revolution: the case of the Cornish pumping engine. *Cambridge Journal of Economics*, *28*(3), 347-363.

Phillips, D. E. (2009). *The software license unveiled: how legislation by license controls software access*. Oxford University Press.

Pisano, G. P. (2015). You need an innovation strategy. *Harvard Business Review*, *93*(6), 44-54.

Pisano, G. P., & Teece, D. J. (2007). How to capture value from innovation: Shaping intellectual property and industry architecture. *California management review*, *50*(1), 278-296.

Posner, R. A. (2005). Intellectual property: The law and economics approach. *Journal of Economic Perspectives*, *19*(2), 57-73.

Rajala, R., Nissilä, J., & Westerlund, M. (2006, June). Determinants of OSS revenue model choices. In *ECIS* (pp. 1839-1850).

Raymond, E. S. (1997). The cathedral and the bazaar.

Romer, P. M. (1990). Endogenous technological change. *Journal of political Economy*, *98*(5, Part 2), S71-S102.

Samuelson, P., Davis, R., Kapor, M.D. & Reichman, J. H. (1994) *Manifesto concerning the Legal Protection of Computer Programs,* Columbia Law Review., Vol. 94:2308.

Schilling, M. A. (2017). *Strategic management of technological innovation*. McGraw-Hill Education, Fifth Edition.

Scotchmer, S. (2004). *Innovation and incentives*. MIT press.

Shemtov, N. (2017). Beyond the Code: Protection of Non-textual Features of Software. Oxford University Press.

Slaughter, S. A. (2014). *A Profile of the Software Industry: Emergence, Ascendance, Risks, and Rewards*. Business Expert Press.

Strowel, A. and Utku, S. *The Trends and Current Practices in the Area of Patentability of Computer Implemented Inventions within the EU and the US,* A study prepared for the European Commission DG Communications Networks, Content & Technology, European Commission, Directorate-General of Communications Networks, Content & Technology.

Teece, D. J. (1986). Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research policy*, *15*(6), 285-305.

Teece, D. J. (2018). Profiting from innovation in the digital economy: Enabling technologies, standards, and licensing models in the wireless world. *Research Policy*, *47*(8), 1367-1387.

Tran, J. L. (2016). Two years after Alice v. CLS Bank. *Journal of the Patent & Trademark Office Society*, *98*, 354.

Young, R. (1999) *Giving It Away: How Red Hat Software Stumbled Across a New Economic Model and Helped Improve an Industry*, in DiBona, Ch., Ockman S., Stone M. ed. Open Sources: Voices from the Open Source Revolution, O'Reilly Media, Inc.

Walden I. *Open source as philosophy, methodology, and commerce. Using law with attitude* in Shemtov, N., & Walden, I. (Eds.). (2013). *Free and Open-Source Software: Policy, Law, and Practice*. Oxford University Press.

Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. O'Reilly Media, Inc..

Winston, E. I. (2006). Why Sell What You Can License-Contracting around Statutory Protection of Intellectual Property. *Geo. Mason L. Rev.*, *14*, 93.

WIPO, Standing Committee on the Law of Trademarks, Industrial Designs and Geographical Indications, Thirty-Sixth Session, Geneva, October 17 to 19, 2016. *Compilation of the replies to the questionnaire on graphical user interface (GUI), Icon and Typeface/Type font designs*.

Välimäki, M. (2005). The rise of open source licensing: a challenge to the use of intellectual property in the software industry. Turre publishing.

Vasudeva, V. N. (2013). A Relook at Sui Generis Software Protection Through the Prism of Multi-Licensing. *The Journal of World Intellectual Property*, *16*(1-2), 87-103.

Analysis was conducted in R (R Core Team, 2019) and figures were produced using the package ggplot2 (Wickham, 2009).

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/

Dowle M, A Srinivasan, T Short, S Lianoglou with contributions from R Saporta and E Antonyan (2015). data.table: Extension of Data.frame. R package version 1.9.6. http://CRAN.Rproject.org/package=data.table

Fox J, Weisberg S (2019). *An R Companion to Applied Regression*, Third edition. Sage, Thousand Oaks CA. https://socialsciences.mcmaster.ca/jfox/Books/Companion/.

Wickham, H (2009). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York

Wickham H (2007). “Reshaping Data with the reshape Package.” *Journal of Statistical Software*, 21(12), 1–20. http://www.jstatsoft.org/v21/i12/.

Wickham H (2011). “The Split-Apply-Combine Strategy for Data Analysis.” *Journal of Statistical Software*, 40(1), 1–29. http://www.jstatsoft.org/v40/i01/.

ANNEX: SURVEY QUESTIONNAIRE

Validation requirements

- Validation rules will be set for the complete questionnaire. Answering all the questions will be compulsory, to avoid partially completed questionnaires.
- Providing information for all the 'please specify' options' is not compulsory.

Useful information for completing the survey:

An OSS application is an application available together with its full source code, under a licence which allows anyone to use it and distribute for any purpose (including commercial use and distribution), both in original and in a modified version (including derivative works).

Standard-form OSS licences are listed at: https://opensource.org/licenses/alphabetical. They contain only a minimal set of obligations, such as attribution or extending the same terms on user modifications (copyleft). Software available without source codes, with fewer permissions, or further obligations (e.g. freeware, shareware) is not OSS, even if it is available free of charge.

For the purpose of the present study we do not consider open-source browsers nor open-source webmail clients.

Section 0. Demographic information

All respondents

1. What was the approximate number of employees in your organisation at the end of 2017? (single choice)
   a) 1
   b) 2-9
   c) 10-49
   d) 50-249
   e) 249+
   f) Do not know

2. What was your total annual turnover (total net sales) in 2017? (single choice)
   a) Up to EUR 500 000

b) Between EUR 500 000 and EUR 2 million

c) Between EUR 2 million and EUR 10 million

d) Between EUR 10 million and EUR 50 million

e) More than EUR 50 million

f) Do not know / will not answer

3. In which sector is the main activity of your business? (single choice + text box)

a) Software publishing

b) Computer programming / software development, programming / IT consultancy and related activities

c) Data processing, hosting and related activities, including web portals

d) Other sector

Please specify _____

4. Which geographic segment of the market generates the majority of your turnover? (single choice + text box)

a) Local

b) Regional

c) National

d) Other EU countries

e) Non-EU countries

Please specify _____

5. What year was your company established? (4-digit number field)

Section 1. Firm's policy in relation to open-source software

All respondents

6. What statements best describe your firm's policy on development and use of OSS applications? Please select only one option. (single choice)

a) My company encourages employees to engage in development and use of OSS during office hours and as a part of their duties	If the answer is 'a)' continues to Section 2: Questions 7, 8 and 9.
b) Use of OSS by employees is actively encouraged by my company, however our employees are not involved in OSS development during office hours	If the answer is 'b)', continues to Section 3: Questions 10 and 11.
c) Use of OSS by employees is allowed if it is the most appropriate tool for the task, however our employees are not involved in OSS development during office hours	If the answer is 'c)', continues to Section 3: Questions 10 and 11.
d) Our employees are not involved in developing OSS nor do they use OSS at work	If the answer is 'd)', continues to Section 4: Question 12.
e) I am not sure	If the answer is 'e)', continues to Section 5: Question 16.

Section 2. Involvement in open-source development

(If the answer to Question 6 is 'a) My company encourages employees to engage in development and use of OSS during office hours and as a part of their duties')

7. What amount of working time do employees spend in the development of open-source projects? (single choice)

a) <25 %

b) 25 %-50 %

c) 50 %-75 %

d) >75 %

e) Do not know

8. How much has working time spent on the development of OSS changed over the last 3 years? (single choice)

a) It has strongly increased (by over 25 %)

b) It has moderately increased (between 0-25 %)

c) It has remained about the same

d) It has moderately decreased (between 0-25 %)

e) It has strongly decreased (by over 25 %)

f) Do not know

9. What are the most important benefits of employee engagement in developing OSS during office hours? Please provide an answer to each statement. (single choice + text box)

	Strongly agree	Tend to agree	Neither agree nor disagree	Disagree	Strongly disagree	Do not have an opinion
It improves their overall programming competences						
It allows us to develop tailor-made solutions in a more efficient manner than proprietary alternatives						

It increases our reputation in the market						
It facilitates entry into new markets						
It decreases the cost of software maintenance and updates due to community crowdsourcing						
It helps to maintain a competitive and level-playing field due to standardisation and common platform building						
Other Please specify _____						

Section 3. Use of open-source products

(If the answer to Question 6 is 'b) Use of OSS by employees is actively encouraged by my company, however our employees are not involved in OSS development during office hours' or 'c) Use of OSS by employees is allowed if it is the most appropriate tool for the task, however our employees are not involved in OSS development during office hours')

10. How much OSS is used in comparison to proprietary software in your company? (single choice)

a) <25 %

b) 25 %-50 %

c) 50 %-75 %

d) >75 %

11. How much has the usage of OSS in your company changed over the last three years? (single choice)

a) It has strongly increased (by over 25 %)

b) It has moderately increased (between 0-25 %)

c) It has remained about the same

d) It has moderately decreased (between 0-25 %)

e) It has strongly decreased (by over 25 %)

f) Do not know

Section 4. Limits to the use or development of open-source products

(If the answer to Question 6 is 'd) Our employees are not involved in developing OSS nor do they use OSS at work')

12. What are the factors limiting your use and / or engagement in development of open-source products? Please provide an answer to each statement. (single choice + text box)

	Strongly agree	Tend to agree	Neither agree nor disagree	Disagree	Strongly disagree	Do not have an opinion
I do not have enough information about the availability of open-source products that could be useful for my company						
Terms of copyright licences of open-source products are not clear enough to me						
It is very difficult to find profitable business models based on open-source						
The open-source governance model does not guarantee continuity of product development which can threaten our business model in the future						
Other Please specify _____						

Section 5. Business models

(If the answer to Question 6 is 'a) My company encourages employees to engage in development and use of OSS during office hours and as a part of their duties', 'b) Use of OSS by employees is actively encouraged by my company, however our employees are not involved in OSS development during office hours', or 'c) Use of OSS by employees is allowed if it is the most appropriate tool for the task, however our employees are not involved in OSS development during office hours'.

If the answer to Question 6 is option a), the user will continue to this section only after Section 2 has been completed.

If the answer to Question 6 is options b) or c), the user will continue to this section only after Section 3 has been completed.)

13. How does your company benefit from involvement in open-source projects? Please select all that apply. (multiple choice)

a) My company distributes open-source products

b) My company provides services to external clients complementing open-source products, such as customised installations, training, help-desk services, etc.

c) My company develops software complementing open-source products

d) My company embeds OSS in tangible products

e) My company uses OSS exclusively for internal processes and we do not offer any product or service derived from open-source products to external clients

f) Other

Please specify _____

14. How important was the existence of open-source products in your decision to start the company? (single choice)

a) It was crucial for the decision to start our business. Without OSS our firm would not have started

b) It was an important factor; however it was not crucial in the decision to start the firm

c) It did not have any impact on the decision to start the firm

15. What types of open-source applications are used / developed in your company? Please select all that apply. (multiple choice + text box).

a) Operating systems

b) OSS development tools

c) Open-source IT security

d) Web server software

e) CMS and / or ERP systems

f) Desktop software (e.g. text processing or spreadsheets)

g) Other types

Please specify _____

16. All respondents. What are the most important considerations when deciding to adopt new software at your company? Please provide an answer to each statement. (single choice + text box)

If the answer to Question 6 is option a), the user will continue to this section only after Sections 2 and 5 have been completed.

If the answer to Question 6 is options b) or c), the user will continue to this section only after Sections 3 and 5 have been completed.

If the answer to Question 6 is option d), the user will continue to this section only after Section 4 has been completed.

If the answer to Question 6 is option e), the user will continue directly to this section.

	Very important	Fairly important	Not important
Total Cost of Ownership (TCO)			
Possibility to freely adapt code to our needs			
Possibility to commercialise the derivative software we create			

Compatibility with software already in use at our company			
General compatibility with software available on the market			
Quality			
Support services available for software			
Other Please specify _____			

17. All respondents. Does your company distribute software developed by your employees? (single choice)

a) Yes
b) No

18. If the answer to Question 17 is 'Yes'. What is the percentage of revenue derived from the sale of licences for software developed in your company? (single choice)

a) <25 %

b) 25 %-50 %

c) 50 %-75 %

d) >75 %

Section 6. Intellectual property aspects

19. If the answer to Question 17 is 'Yes'. What are the licence types you use to distribute the software created in your company? Please provide an answer to each statement. (single choice + text box)

	Very often	Often	Sometimes	Seldom	Never
Proprietary licences					
Based on BSD licence					
Based on MIT licence					
Based on Apache licence					
Based on Mozilla Public licence (MPL)					
Based on Lesser GPL (LGPL)					
Based on General Public licence (GNU)					
Other Please specify _____					

20. All respondents. How do you rate the effectiveness of various measures of protection for your products and / or services? Please provide an answer to each statement. (single choice)

	Highly effective	Moderately effective	Not effective at all	I do not use it
Trade mark				
Patent				
Copyright				
Design				

Utility model				
Database law				
Confidentiality (trade secret)				
Complexity of products or services				
Leveraging complementary assets (production, implementation or marketing capabilities of my firm)				
Time to market				
Internet domain name				

Section 7. Perception of open-source

21. All respondents. What is your company's perception of open-source products? (single choice)

a) Mainly an opportunity for my current business model

b) Both a threat and an opportunity depending on the product / service line

c) Mainly a threat for my current business model

d) It does not affect my business model at all

22. All respondents. Please, provide an answer to each statement. (single choice)

	Strongly agree	Tend to agree	Neither agree nor disagree	Disagree	Strongly disagree	Do not have an opinion
Open-source products offer better quality than proprietary alternatives						

Open-source products are more reliable than proprietary alternatives						
Overall, the cost of open-source products (including initial training and maintenance) is lower than the cost of proprietary alternatives						
Open-source products increase market competition by lowering the entry barriers and enhancing the possibility of software firms finding profitable business models						
Open-source significantly broadens the choice of suppliers and quality of software						
Open-source seriously limits the possibility of software firms to find profitable business models						
The current system of IPR protection is beneficial to my firm						

OPEN SOURCE SOFTWARE IN THE EUROPEAN UNION

June 2020

ISBN 978-92-9156-276-3 doi: 10.2814/866548 TB-03-20-304-EN-N