Journal of Business and Trade Economics | **Volume: 06 / N°: 01 (2021), p 123-133**

Towards a new form of free merchandising with Open Source Software

Nadia Medrous[1], Khadidja Nemmiche[2*]

[1]University Centre of Maghnia (Algeria), khadidjanemmiche@gmail.com
[2]University Centre of Maghnia (Algeria), nadiamede@yahoo.fr

Received: 23-09-2020 Accepted: 01-03-2021 Published: 31-03-2021

Abstract:

In this paper, one of the most recent phenomena keeping pace with the introduction of the Internet is discussed. The free market and the free use of software is a new trend that is upsetting the thinking that has been going on for a long time. Choosing to purchase or download open source software and then restructure its functionalities according to the user's own needs, or to participate only in a virtual embedding, means a new orientation towards economic and social relations different from those known until recently. Using a conceptual approach, the results of this research suggest that free software creates a new virtual and specific market relationship that broadens the scope of freedom given to users.

Keywords: *Free software, Merchandising, Application market, Community, Internet.*
Jel Classification Codes**:** L17, L24.

1. INTRODUCTION

The digitalization of economic and social activity; a new paradigm presented by Horn (2004) to illustrate the undoubted importance of the use of software in different fields. The adoption of application tools is today an essential element on which human behavior is based. Both for the user firm and for a simple social consumer, software has become an inseparable part of human life. Acquiring software is for a long time a problem largely posed by the end user. For a client firm, software is considered an indispensable tool through which the various functions are carried out. From an economic and managerial point of view, the question of application acquisition remains a crucial analytical issue, leading the user firm to redefine its action plan beforehand.

Streamlining the chosen approach requires weighing all possible solutions from different angles. Adopting merchant software is one of the most frequently selected choices. For a while, the market remained the most preferred solution for professional use by firms. In the application market, the offer is extremely diversified. Intense competition and strong technological obsolescence enrich the market extension in terms of application satisfaction.

[*] ***Corresponding author***

Every day, a new form of software emerges that corrects complements or overwrites the previous version.

Today, a new software fragmentation has emerged that gives an extremely broad framework of freedom not only for the professional use of software, but also for social use. This is open source software that may have produced a new direction of merchandising for traditionally recognized products and services.

In what follows, our pioneering attempt aims at presenting for the first time in the economic and managerial literature the new market structure of free software. In reality, the characteristics of free software reflect a particular form of market, consumption and exploitation for a firm seeking to acquire free software to satisfy its own application needs.

Indeed, the problem posed takes the following form:

What kind of change has occurred with the introduction of free software in economic life?

To answer this question of analysis, the following hypothesis is proposed:

Free software has created unmistakable change from different angles after expanding the scope of application freedom given to users.

The conceptual approach adopted in this paper will allow us to focus on the real impact of open source software on the technological behavior of economic actors.

2. Software from a managerial point of view:

Recently introduced into managerial, economic and social life, software, after having been considered as an object of study that is exclusively part of software engineering, has in recent decades become an important subject of analysis in various disciplines without being restricted to a purely technical framework.

The French word *logiciel* was first proposed in 1967 by *P. Renard* to replace the English term *Software* which belongs to the non-tangible part of the computer. The software is defined by Haigh (2002, p. 6) as a written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory. In its overall sense, the software refers to all the programs essential to the execution of computer systems. It is seen as a message or set of instructions from man to machine or as an intermediary between the user and the computer that aims to translate the problems of human actors into the machine's repertoire (Horn, 2000).

As a computer product, software is considered a text written by a programmer developed by a particular programming language in the form of programming lines; called source code, software source, or program source. This code is compiled and automatically transformed into a binary suite called *machine-code*, *object-code*, *binary* or *executable* (Caillaud, 2003, cited by Blum, 2007). At the managerial level, software is part of the computerized information system; a system that is based mainly on the use of technology and computers to operate. Acquiring software is made possible through an organizational trilogy forming what is called by the choice of Sourcing. The aim is to select from among three

solutions the one that is appropriate in relation to the expectations of the client firm (Williamson, 1991).

The first solution amounts to an application self-satisfaction called *Insourcing*, whereby the firm entrusts the software development to its internal IT department in order to meet a requirement that has appeared in the past (Alavi and Carlson, 1992). Generally, this trend is preferable when the software product is highly specific to the firm requesting it. Developing software based on in-house skills is considered an expensive but sometimes irreplaceable choice when the other two organizational strategies fail to meet certain needs or are perilous in the long term.

However, the second option refers to cooperation with another legally independent party. This approach is known as *Outsourcing* and refers to the fact of entrusting the production of a given piece of software to a third party. The Outsourcing of software development to a specialized external provider is one of the most significant trends in business today. Compared to the client firm, the provider is better able to carry out a software development project in terms of quality and costs. By entering into a medium or long-term Outsourcing contract with the selected service provider, the firm is relieved of this managerial burden and concentrates on functions that are supposed to be more strategic.

As a third solution, the market is seen as the simplest and most traditional choice. Compared to the other two organizational substitutes, the market offers standard software products designed to meet the needs of anonymous parties. Since the application production of generic functionalities at the producing party is done on a large scale, economies of scale are easily achievable and the price of such offers will therefore be attractive. For a client firm, the software packages that computerize the firm's internal management are diverse, serving multiple intra-functional and intra-sectoral needs (Son et al, 2014).

3. The Birth of Software Merchandising

It has to be said that the merchandising of software came about after the concrete separation of hardware and software, which previously constituted a single technical component in the form of an inseparable computer system. Two elements made computer fragmentation at that time impractical. This is the extremely heavy burden of using IT in the firm and the simplicity of the needs expressed by users.

Until 1960, there were several thousand computers in circulation, 90% of which were owned by IBM alone (Carmel, 1997). The very high costs of this new technology had led large firms and administrations to allocate technical components instead of buying them. Thus, each computer manufacturer had its own technical infrastructure while ensuring endogenous accounting between its machines and applications. In addition, the IT needs in its early stages were very simple in terms of their formulations and requests. Contrary to what is seen today, development costs were only 3% of total IT costs because the software was very simple with limited functionality. This has identified the market independence of applications from the hardware on which they are installed.

It was not until 1970 that the separate merchandising of technical components and applications was introduced. Under pressure from antitrust law in the *United States,* IBM adopted a practice of separate billing (*Unbundling*) between hardware and software in the early 1970s. Separate billing of each of these components by IBM was essential when corporate customers began to demand increasingly complex requirements.

Just after IBM's pioneering strategy, the IT market experienced a remarkable boom. The high profits generated by computer manufacturers led to increased competition between those offering approximately similar services. Thus, the potential for technological obsolescence due to the acceleration of the software life cycle was very high. In addition, corporate clients have begun to demand increasingly complex and diverse functionality. It was therefore essential for manufacturers to focus on a single activity by freeing software production from hardware construction.

Within the IT market, technical and application standardization has become mandatory, leading to the emergence of independent firms focusing on a particular segment. Development was a main, recurring and large-scale activity for these publishing firms. These firms, which initially sought to increase their market share, produce and develop standard and generic software. For publishing firms, the standardization of the production process, which targets generic and anonymous customer needs, makes it easy to achieve significant economies of scale. In addition, software production costs are incurred only once during the primary development phase. The costs of any subsequent duplication are disregarded.

For firm customers, commercial software is a more favorable choice compared to in-house and Outsourcing, particularly in terms of cost. The pre-programming and pre-determination of functionalities make the adoption of such solutions immediate. Thus, the economies of scale achieved by the producing parties are positively reflected in the price of their offers. The merchandising of software takes two forms depending on the nature under which the software is acquired. Software can be considered as a product just as it can be considered as a service. In this framework, Horn (2004) sees it as a product if it is burned on a storage device. However, the ability of the user to download the software via the Internet with or without a cost makes it possible to present the software as a service.

4. The software package: a commercial product

Producing and selling generic software in the market to satisfy the needs of anonymous parties is the strategy adopted by software firms to generate extraordinary profits. These firms most often focus on what are known as software packages.

Literally, a software package is presented by the conjunction of two words: product and software. A software package, also called *Packaged Software, Off-the-shelf-Products* or *Off-the-Shelf-Product,* is an application developed in a standard form by a software firm and with features general enough to be used by a large number of end users. According to Drean (1996, p. 68): "*The software package is a coherent and independent set of programs, services, information handling media, docket information, languages, etc. and documentation designed to carry out standard computer processing operations, the distribution of which is of a*

commercial nature and which a user can use autonomously after a full implementation and training".

The birth of the merchandising of software packages has created a new technological and economic era where publishing firms found in application segmentation a perfect asset to generate huge profits. The software packages began to be offered in large quantities to a number of anonymous clients. In the software package market, two forms of products are distinguished according to the functionalities they fulfill. According to Jadhav and Sonar (2009), there are software packages exclusively for business use, also referred to as software for professional use and software packages for the mass market.

As for the first form, these software packages are intended to be used by client firms to support its various functions performed within its borders. Among these, there are those that are aimed at a unique and particular sector of activity (e. g. finance, pharmacy, etc.) and those intended for cross-sectoral activities and concerning a particular function of the firm (accounting, human resources management, supply management, etc.). In addition, enterprise resource planning (ERP) systems are designed to cover the entire firm's activities.

On the other hand, consumer or public domain software packages are consumer goods intended for the general public or simply for microcomputer users. These software packages are sold in large numbers at a lower price compared to software packages for professional use. This form of software package includes system infrastructure software (e. g. operating systems, middleware, system management software, and security software), software development tools (e. g. database management systems, components, objects and development environments, development lifecycle management and Internet tools).

5. Closed source software: a restrictive solution

One of the criteria used to categorize commercial software is the accessibility of the source code. In this context, two families of software are distinguished. They are both closed source and open source software. The free use of software or its shrinkage has been a subject of technological, economic and managerial analysis widely adopted by researchers over the last two decades. Proprietary software also called non-free or closed source software is a software product where the publisher sells only the use of the product. Its redistribution or modification remains prohibited. This software can be classified as: commercial software, shareware and freeware.

As for commercial software, Jadhav and Sonar (2009, p. 556) define it as follows: "Commercial software is software which is purchased through the retail market";. To use commercial software, the customer must make the purchase, as such software is not available free of charge. The licenses for the use of this software prohibit any access to the source code, any kind of duplication or redistribution. The customer can only use the functionality of the software in its executable form. However, shareware is software that is distributed free of charge, possibly for a test period only. However, these programs offer only a limited number of features. Replaying or modifying such programs is not permitted. After the trial period, the user must purchase the full version of the software if he is satisfied with the software and wants to use it even more.

For its part, a freeware, as its name indicates, is offered free of charge regardless of its user license. The source code of the program is not available, which therefore prohibits any kind of modification. Although freeware is very similar to shareware, the main difference lies in the price. These programs include limited functionality that is provided free of charge for an unspecified period of time.

In the proprietary software market, copyright is retained as long as the simple reading of the code reveals the algorithms and functionalities. Faced with an evolution in the number of programmers, program transcription allows anyone to market a software product already sold by the original publisher. Indeed, and like any intellectual product, the protection of property rights is necessary to guarantee and reserve the rights to the original developers. The non-disclosure of source code and the implementation of restrictive user licenses help protect this commercial asset. For this reason, this software is distributed only in executable form, according to which the content and the internal working mode remains a secret by its original publisher. The user cannot therefore decompile or modify the code, which is provided exclusively in binary.

Admittedly, this form of merchandising is quite limited in narrowing the perception of satisfaction with the software. Whatever the sphere of use, the client firm or a simple consumer for non-professional use is faced either with functional under-use which makes the software deficient, or with over-use which makes the pre-established technological infrastructure more cumbersome. In fact, this case can lead to a market failure for the closed software. 5. From closed source to open source software For the client firm, closed source software was for a while a commercial solution that was assimilated in terms of its organization to other products available on the market. The restrictive framework of the freedom given to users has had a negative influence on market application satisfaction.

The new technological trend towards more freedom for end users to use software is seen as an extraordinary opportunity for the client firm to almost completely control its application needs. For the latter, the new market segmentation of free software creates a new form of free software that is extremely different from what was previously recognized.

Free software also known as Open Source software is characterized by a disclosure of its source code to allow users to make changes after obtaining the software. Free software thus freely discloses the source code and is defined by Foray and Zimmermann (2001, pp. 81-82) as: "software whose source code, i. e. the set of instructions that form the program before compilation, is made openly available and cannot be appropriated privately";. The purpose of this software is to let the user restructure the developed functionalities in order to make the product completely adequate with the real requirements of the user party.

For Richard Stallman, the founder of this new generation of software, private property is guaranteed by the implementation of certain special licenses to protect the specific nature of such software (e. g. the General Public License GPL, the Berkeley Public License BPL). Public licenses confer four rights to users of open source software, but require any modified version to be protected by the same license, thus prohibiting conventional license repositories, i. e. make free software proprietary software (Dedrick and West, 2005).

The term free software sometimes has a confusing meaning because for the English-speaking world free means free, but also free. In reality, this software is called free software because access to the source code is free, with no relation to price. Therefore, free software is not a consequence of the freedom of the source code. As there is free open source software, there is also such software that is paid for use (Dedrick and West, 2005). Today, there is a growing substitutability of closed source software by applications that give more freedom to the user. Open source software makes it possible to reorient its functionalities according to the end user's purpose (Dalle and Jullien, 2001). The flexibility produced by the adoption of such software has led to strong competition with traditional software, i. e. closed source software.

6. New consumption particularly free

In contrast to proprietary software with veiled source code, Open Source software is characterized above all by the availability of the source code. The possibility for each user to modify and improve it is made possible. Free software freely discloses source code. It is therefore a question of the user, not the developer, to redefine its functionalities according to its real needs. In addition to the publicity of the source code, the free software license guarantees four freedoms, called the fundamental freedoms of free software, mentioned as follows: The freedom to run the program for any kind of use. The freedom to study how the program works and adapt it to needs. The freedom to redistribute copies. The freedom to improve the program and to publish improvements without being obliged to communicate on this subject with the developer or any other specific entity.

Certainly, these four freedoms have tilted the favour towards this new form of software, representing today a strong competitor to closed source software that is characterised by a predetermined and limited scope of application. The adoption of such an open-source solution leads to the functionalities being alongside the needs expressed by the users, since it is they themselves who define the desirable evolutions. This choice allows the user to use the acquired software as it is, to redirect its functionalities according to his requirements and according to the pre-established technical and application structure, or to make modifications and distribute them.

Moreover, free software has a sometimes low cost. It can often be obtained free of charge since it can be downloaded from the Internet without charge. This reduces costs because; instead of developing new software by exploiting specialized project resources, only a few modifications to the software are required to make it suitable for internal needs. Kenwood (2001) points out that FOSS provides user firms with a better price/performance ratio. The quality of Open Source software can be superior to that of proprietary software. This software is less prone to piracy because opening the code allows several people to identify and correct defects. Den Besten et al (2008) cite that open source software is relatively less voluminous and requires fewer resources to operate effectively.

Opening the source code does not mean that this software is any less secure. Open source software has been designed for use in a multi-user environment and security features have been well adapted to this function (Lerner and Tirole, 2002). A study conducted by MITRE (2002) on the major elements of US Department of Defense security indicates that

developers of such software ensure the confidentiality of the user party's data. The fact that the code is freely available allows hackers to see weaknesses in the software; but conversely, programmers can quickly identify problems and flaws in the software. When a bug is discovered, it is quickly fixed and a software update is performed very quickly.

Improvements and innovations are always accumulated in the most recent version of the software and everyone can use it. Thus, open source software has a longer lifespan because the accessibility of the source code makes it easy to carry out maintenance that extends and updates the software (Mangolte, 2006).

7. The Open Source Software Market

Today, the great technological, economic and managerial prosperity of Open Source is threatening the position of proprietary software which until recently dominated the software market. The huge success of some open source software such as Linux, Apache, PHP and PERL and the clear preference of several firms to adopt an open source strategy at the expense of closed source software make the open source market a rapidly growing market.

In a study conducted in 2013, the Forrester firm lists the rates of use of Open Source software by European firms. 79% of the firms surveyed were using Open Source software bricks in the web infrastructure, 67% of the firms were using Open Source software in development tools and 59% in critical application tools such as databases. For example, 21% of firms used Open Source software for applications.

The Open Source market is expected to reach nearly $8. 1 billion in 2016, with a growth rate of 22. 4% per year. This market amounted to 12. 68 billion dollars in 2018 and will reach 66. 05 billion dollars in 2026 according to Verified Market Research (2019), with a growth rate of 22. 94% expected between 2019 and 2026. In the following table, the utilization rate of OSS is classified by country, industry and function, respectively.

Table 1: Ranking of Open Source usage

Country	North America: United States; Canada; Mexico. Europe: Germany; Brittany; France; The rest of Europe. Asia Pacific: China; Japan; India; The rest of Asia Pacific. Latin America. The Middle East and Africa.
Sector	Banking, financial services and insurance, Health services, Industry, Detailing and distribution, Others.
Function	Support, maintenance and service management, Training Consultation, Implementation.

Source: According to the authors, depending on the statistics of Verified Market Research (2019)

One of the markets where Open Source software dominates is that of web servers, with in 2015, 64% of the market share for Apache, 29% for Microsoft and 7% for the others. The American firm Red Hat is the leader in this market, with sales of around $120 million in fiscal year 2015. The Linux operating system alone accounts for 35% of the operating system

market and its commercial versions are growing at an annual rate of 25%, higher than that of other operating systems. Thus, Sendmail for mail servers alone holds 80% of the mail traffic.

The profits held by these firms do not lie in the sale of free software but in the marketing of services and activities that affirm them such as: software selection, copying, distribution, warranty, maintenance, integration, consulting, installation, technical assistance, development of specific solutions, etc. Open Source software does not directly generate significant added value, but only associated IT departments or the development of proprietary code based on open code can create it. The success of Apache is a good example of this. The software is free but the added value of the seller is mainly on the sale of services (specification and implementation of the web server, hosting, facilities management) (Audris et al., 2000). A person interested only in the software can download it for free and compile it from its source code which is directly accessible.

8. The Free Software Community: A New Business Relationship

It is fair to say that the increase in the market share of FOSS reflects the technological power of FOSS. According to several researchers, the key factor of success that allows them to crush their competitors is the disclosure of source code, which has formulated a virtual embedding that supervises several developers. This new cooperative trend has established a consortium that initially seeks to increase the application quality of open source software to be used. By freeing up the use of software, developers begin to share a body of knowledge, experience and support with third parties across the Internet for the purpose of correcting faults or perfecting pre-developed software (Dempsey et al, 2002).

The Internet is the raison d'être of the free software community, which brings together passionate people, often of different nationalities, connected only by a computer network and welcoming any new participant. According to several authors, the superior quality of Open Source software is mainly due to the activism of the community and the high level of developer skills it understands.

On the other hand, the concept of the free software community does not limit the development of this form of production to exclusively collective work. There are no clauses in open source licenses that require cooperation with others in the production and development of a free program or more generally that define how the code should be made. Producing free software can be a lonely job. This situation is very frequent and takes the name of the model of the cellar. A single programmer writes his source code, compiles it, tests it, and makes it available under a special license to everyone, including any bug fixes that are sent to him. In his 2002 study, Krishnamurty suggests that more than 50% of the active and mature projects in the Sourceforge database were carried out by a single individual, while others involved larger teams.

The author reports that 29% of projects had involved more than 5 developers (Krishnamurty, 2002). On the other hand, members of the free software community cannot necessarily be volunteers. Although a large number of volunteers help to keep the FOSS movement alive, a considerable proportion of developers receive direct or indirect compensation (Benkeltoum 2009). According to Lakhani and von Hippel (2000) about 20%

of the programmers contributing to FOSS development are paid to do so, the others are volunteers, most of them devoting an average of a few hours a week to this activity. For example, 70% of the modifications made to the Linux kernel (the core of the GNU/Linux operating system) are attributable to paid firms (West, 2003).

The market form of the free software community today represents an object of study enriching the managerial literature on the question of the firm and the movement of its boundaries. Within this framework, this community forms a new and specific governance structure that lies between hierarchy and market and does not return to the previously recognized hybrid form. Admittedly, the extremely specific nature of relations between community players on the one hand, and between developers and users on the other, points to a new organisational form which belongs neither to strictly commercial relations nor to voluntary subordination, nor even to a structure of cooperation between two economic parties.

It is therefore a new relational image between a cluster offering its application services via the Internet, whether for profit or not, and a cluster receiving these applications to satisfy its application needs (Nemmiche et al, 2014). This satisfaction may concern the needs of the user himself who accesses free software or simply to become active in the community through a partial modification of the source code with the aim of re-distributing it. Indeed, the free software community deserves to be seen as a new organizational conception that is born between independent parties connecting in the virtual world whose goal is to start inventing or innovating in order to share with others the success and technical completeness of the software.

9. CONCLUSION

Today, software is considered an indispensable element to run and support the different activities of a firm. Application needs are increasingly diverse and the end user demands complex and branching functionalities. The question of functional satisfaction thus becomes an issue for the client firm. Closed source software available on the market cannot therefore fill the gap between what is expressed by users and what is available. Such a unilaterally predetermined generic service with no established relationship between developer and end user narrows the predicted level of satisfaction.

It thus appears that FOSS has produced a radical change that affects different angles in the life of firms and the simple consumer. By disclosing the source code of the program, free software has truly expanded the scope of freedom given to users. This freedom allows resizing the product according to the real needs without confronting either the under-use of the software or its over-use.

With open source software, a new, more flexible market segment has been created through which the user reorients the functions included according to his real requirements. Whatever the nature of the use of the software, the customer finds in the use of open source solutions a strategy more beneficial and closer to the needs expressed. Open source software represents a new market structure in which its commoditization differs from the traditional form previously recognized. Compared to other commercial services, Open Source software

formulates a specific new conception of its merchandising. Thus, the virtual relationship that links the developer with the software user illustrates an incomparable cooperative structure.

Certainly, free software has given rise to a new consideration to the economic organization deployed for a long time; for a new form of commoditization is established between two independent poles and a new interdependent bipolar relationship encases the programmer and the end user.

10. Bibliography List:

ALAVI M., CARLSON P., "A review of MIS research and disciplinary development", *Journal of Management Information Systems*, Vol.8, n.4, 1992, pp. 45-62.

AUDRIS M., FIELDING R., Herbsleb J. "A Case Study of Open Source Software Development: The Apache Server", *Proceedings of the 22nd International Conference on Software Engineering*, ACM, 2000, pp.263-272.

BENKELTOUM N. "Les régimes de l'Open Source: solidarité, innovation et modèle d'affaire", *Thèse de doctorat en sciences de gestion*, Université de Mines Paris Tech, 2009.

CAMPBELL-KELLY M., GARCIA-SWARTZ D.D, "Pragmatism, not ideology: Historical perspectives on IBM's adoption of open-source software", *Information Economics and Policy*, Vol.21, 2009, pp.229- 244.

DALLE J.M., JULLIEN N. "Libre'Software Turning Fats into institutions?", 2001

DEMPSEY B.J., WEISS D., Jones P., Greenberg J. "Who is an Open Source Software Developer", *Communications of the ACM*, Vol.45, n.2, 2002, pp.67- 72.

DEN BESTEN M., DALLE J.M., GALIA F., The allocation of collaborative efforts in open source software", *Information Economics and Policy*, Vol.20, 2008, pp.316-322.

FORAY A., ZIMMERMANN B. "L'économie du logiciel libre : organisation coopérative et incitation à l'innovation", *Revue Economique*, n.52, Numéro hors série sur l'économie d'Internet, Edité par Brousseau E. et Carrien N., 2001, Octobre.

KRISHNAMURTHY S. "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects", *First Monday*, Vol.7, n.6, 2002, pp.47-59.

LERNER, J., TIROLE, J. "Some Simple Economics of Open Source", *Journal of Industrial Economics*, Vol.50, 2002, pp.197–234.

MANGOLTE P., "Marchand et non-marchand dans l'économie des logiciels", *Colloque international, Mutations des industries de la culture, de l'information et de la communication*, Septembre, 2006.

SON I., LEE D. LEE J., CHANG Y.B. "Market perception on cloud computing initiatives in organizations: An extended resource-based view", *Information & Management*, Vol.51, 2014, pp. 653-669.

WEST J."How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies", *Research Policy*, Vol.32, 2003, pp.1259-1285.

WILLIAMSON O., "Comparative Economic Organization: The Analysis of Discrete Structural Alternatives", *Administrative Science Quarterly*, Vol.36, 1991, pp. 269-296.

Verified Market Research (2019), consulted on: 15/02/2020. https://www.verifiedmarketresearch.com/product/open-source-services-market/