Open Source Software (OSS) Adoption in Commercial Firms: A Literature Review

Bundit Thanasopon

Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok

Email: bundit@it.kmitl.ac.th

ABSTRACT

This paper reviews previous research on "Open Source Software" (OSS) adoption in two types of organizations, namely software-intensive and non-software-intensive commercial companies. In addition to a brief overview of OSS, the paper presents 7 ways of how software-intensive firms employ OSS products and practices. The other important finding is an identification of 14 key factors influencing OSS adoption decisions in non-software-intensive companies. Some are suggested to encourage OSS adoption, while others are either inhibitors or inconclusive in their effects. The findings could be of interest to managers in both types of organizations and academia alike. Ultimately, several interesting avenues for future research are recommended.

Keywords – Open Source Software; Use of Open Source Software in Firms; Factors Affecting Adoption Decisions;

1. Introduction

Over the last decade, the Open Source Software (OSS) phenomenon has attracted increasing attention from not only the software industry, but also organizations in other sectors [1, 2]. OSS refers to computer software whose source code is freely accessible allowing users to run, copy, distribute, and modify the software [3]. Unlike proprietary software (e.g., Windows 8, iOS), the development of OSS (e.g., Linux, Android) occurs in a public, collaborative manner over the internet. OSS development is typically overseen by groups of core developers and supported by large communities of contributors [4].

OSS proponents often claim that the collaborative nature of OSS development leads to better software quality and reliability, less development time, lower acquisition costs, greater flexibility due to source code accessibility, and freedom from vendor lock-in [3, 5-7]. On the other hand, some have argued that the majority of OSS products, highly visible IS applications in vertical domains in particular (e.g., office applications, ERP, etc.), are still immature and unreliable [5, 8].

Additional disadvantages of OSS are limited documentation and external support, version proliferation, and constraints imposed by the various OSS license terms [5-9].

Organizations define OSS differently depending on the context of the individual organization [10]. Therefore, understanding the conditions under which OSS will be adopted or will not be adopted in specific contexts (i.e., software-intensive versus non-software intensive [11, 12]) could help managers to create a specific organizational environment where OSS can flourish in their organization. To this end, the main objectives of this paper are to: (1) provide an overview of what we know about OSS – its strengths and weaknesses, (2) to understand how software-intensive firms use OSS products and practices and (3) to identify facilitators and inhibitors affecting decision makers in deciding to adopt or not to adopt OSS as end users. The first objective focuses on catering for audiences with limited background on OSS in general. Regarding the second objective, this paper wants to explore applications of OSS products and practices that commonly adopted in product/service development process in software-intensive companies. Finally, since OSS could also benefit commercial firms who are not in the software industry, the author aims to encourage managers in those firms to adopt OSS as end users by identifying factors affecting their adopting decisions from the vast body of literature.

To achieve the three research objectives, the author conducted a comprehensive review of the relevant literature. The review process started by generating a list of relevant journals and conferences based on previous reviews on OSS and OSS adoption. Then, the author searched online digital libraries for publications in the relevant journals and conferences with two keywords, i.e., "open source" and "free software" [11]. The matched papers were individually reviewed by the author based on their title and abstract. Those found to be relevant were then reviewed in full.

The rest of the paper will be organized in the following way. The paper commences with a brief introduction to OSS and its development process. Next, a review of the current research on OSS adoption is presented. After that, the author discusses on what has been found in the literature. Ultimately, conclusion and suggestions for future research are provided.

2. Open Source Software (OSS)

In the early days of electronic computing, when researchers began to use computers for their work, they often had to share source code because the variety and availability of commercial software were very limited [13]. Also, it was practical and financially reasonable for programmers to share their source code among very few of them [12]. As computers became better and cheaper, the number of developers increased, and the source code became more complex. This made sharing free software flourished especially in the academic environments [12]. An example was Barkley Software Distribution (BSD) resulting from the collaboration between the University of California Berkley and AT&T labs as described in [14].

However, since the beginning of 1980s, the idea of close-sourced software became the standard of the industry, taking over the concept of free source code sharing [12]. OSS supporters went to found their own organizations as an alternative to the mainstream proprietary software. In the mid-1980s, Richard Stallman founded the Free Software Foundation. The organization defined the key concepts of OSS saying that the access to computer software source codes should be unrestricted. Anyone should be able to use, modify and circulate the source codes at no costs [15]. More importantly, the Free Software Foundation introduced the "General Public License" (GPL), a legal paper assuring that once software was published under the GPL license, it source code is guaranteed to be freely available for modifications [13]. The Linux kernel is an example of successful software licensed under GPL. There are also other various open source licenses. Essentially, open source licenses differ on for key aspects [3]:

(1) whether they allow OSS to be mixed with proprietary software,
(2) whether modifications can be made private and not returned to the author(s),
(3) whether the software can be licensed by anyone, and
(4) whether the program contains special privileges for the original copyright holder over user's modifications.

In the past 10 years, several large commercial companies have adopted OSS products and entered the OSS arena by initiating OSS projects themselves. For example, Google runs it entire search operation on Linux [3]. Moreover, one of its flagship products – i.e., Android – uses open source licenses [16]. Nevertheless, the adoption rates of OSS are still lagging behind its closed-source counterparts in many software sectors [17, 18]. To gain a better understanding of OSS, the author reviews prior works that discuss on strengths and weaknesses of OSS as well as briefly describe the development process of OSS.

2.1. Strengths and Weaknesses of OSS

The development of OSS has been describes as a "bazaar-like" activity driven by enthusiastic programmers who voluntarily work on the OSS project [14]. The bazaar development style creates an inspiring, creative and democratic atmosphere [14]. Depending on the license, OSS is generally free of charge for private and commercial use [8]. In addition, OSS proponents argue that public scrutiny enabled by the open source structure can result in a product with fewer defects compared to commercial software [11]. Unlike in proprietary software development, typically, there are no constraints on the underlying operating platform; OSS can be applied to most operating systems and environments. Thus, vendor lock-in could be avoided [8]. Finally, as the source code is freely available, skilled users may freely explore and modify the source code to fit their requirements [9].

With regard to weaknesses, critics of OSS claim that ambiguous leadership of OSS projects weakens software quality [9]. Regarding the cost advantage of OSS, while many studies [1, 3, 19] indicated that lower cost helps drive OSS adoption, other have

argued that OSS is not free and might not be less expensive than proprietary software [5, 8]. Also, most users do not find source code accessibility beneficial since very few have ever made any modification to the source code [20]. Furthermore, OSS is developed by independent developers or groups of developers which may have different opinions and ideas on how the project should proceed. Therefore, they may create their own version of software which could exacerbate into a phenomenon called "forking" resulting in two or more different evolutionary paths of the software [3, 5]. The availability of external technical support of OSS products is still far behind their proprietary counterparts. The support of much OSS relies on e-mail list/bulletin boards. Experienced consultancy firms who can professionally perform installation, configuration, and maintenance are still lacking [7, 8].

Table 1. Strengths and weaknesses of OSS

Strengths	Weaknesses
Availability of source code [8, 9]	Version proliferation [3, 5]
Free of charge [3, 8]	Switching costs can be high [2, 5, 9, 21]
Better quality because of the bazaar development style [3, 11, 14]	Ambiguous leadership of OSS projects weakens software quality [5, 9]
Vendor lock-in avoidance [8]	Lack of external support [7-9, 20]

2.2. OSS Development Process

The traditional Software Development Life Cycle (SDLC) in its most generic form has four main phases: planning, analysis, design, and implementation [7]. According to [7], in an OSS project, the first three phases (i.e., planning, analysis, and design) are usually executed by a single developer or a small group of developers. The planning phase starts when a developer perceives a unique problem, thus limited commercial options. He or she, then, tries to develop a solution for that problem leading to an initial prototype. The requirement analysis phase is, however, largely superseded. This is because requirements are taken as generally understood by the developers as they are also the users of the software. Design decisions are likely to be made in advance by the core group of developers who initiate the project. Finally, the implementation phase of OSS projects comprises six subphases. They are: (1) code, (2) review, (3) pre-commit test, (4) development release, (5) parallel debugging, and (6) production release [22].

3. OSS Adoption

Prior works on OSS adoption in commercial firms can broadly be classified into two main research streams: (1) how software-intensive firms adopt OSS products or practices, and (2) what are the factors that facilitate or inhibit the adoption OSS as end users in non-software-intensive organizations. In this paper, the author defines "software-intensive firms" as private or public firms who extensively use or develop software for the provision of their core products or services. On the

other hand, "non-software-intensive firms" are those who use or develop software mainly to support their daily operations or computerized their business processes.

3.1. How OSS is adopted in software-intensive firms

Here is some recent research in the first stream. Hauge and colleagues [11] reviewed the literature and developed a classification framework for organizational adoption of OSS. The framework suggests two main usages of OSS in organizations. Firstly, firms use OSS products in their operational environment as end users. Secondly, firms also use OSS in their software development, which can be classified into five categories: (1) using OSS CASE tools to development their software, (2) integrating OSS components into their software, (3) participating in other's OSS projects, (4) initiating their own OSS projects, and (5) using software development practices commonly employed in OSS development [11].

Similarly, based on interviews and survey data, Hauge [23] defined four industrial OSS roles: (1) OSS providers who release their proprietary products as OSS, (2) OSS integrators who include OSS components into their software products, (3) OSS participants who provide occasional bug fixes and requirements, subscribe to mailing-lists and mainly use the software (4) inner source software participants who use software development practices often used in the development of OSS.

A systematic review of the literature on OSS adoption conducted by [12] has identified four categories of how commercial firms in the software industry use OSS. The four categories are: (1) participating in OSS development communities, (2) building business models around OSS products, (3) including OSS components into their products, and (4) using OSS practices within the firms. The second category is unique from the classification framework proposed by [11]. To be more specific, companies can build business models around an OSS product by offering customized software based on the OSS product, or providing consulting and training services.

Ziemer [10] conducted a grounded theory case study of an industry-driven R&D project and found that the adoption of OSS in industrial software development can be categorized into: (1) developing with OSS tools and practices, (2) developing with OSS components and (3) developing OSS products. Table 2 summarizes the ways software-intensive companies use OSS.

Table 2 Summary of how firms adopt OSS.

Ways of adopting OSS	Ref.
Deploying OSS products in their operation environment as end users.	[11]
Using OSS CASE tools in software development.	[10, 11]
Integrating OSS components into their own software systems.	[10-12, 23]
Participating in the development of OSS products controlled by	[11, 12, 23]

another organization or community.	
Providing their own OSS products and relating to a community around these products.	[10, 11, 23, 24]
Using open source process within a company or consortium of companies.	[10-12, 23]
Building business models around open source products.	[12]

From the review, the author found that, while software- intensive organizations employ OSS products and practices in both their supportive business processes and their core business processes; non- software- intensive firms, on the other hand, mainly adopt OSS products to support their routine operations as end users. In the next section, the paper turns its focus to factors influencing OSS adoption decisions in non-software-intensive firms.

3.2. What factors facilitate or inhibit OSS adoption in commercial firms

Research in the second stream tends to focus on OSS adoption for operation environment as end users in non- software- intensive firms. In other words, it tries to identify facilitators and inhibitors of the diffusion of OSS in firms in industries other than the software industry. There are various ICT adoption models and theories that have often been used by scholars in this research stream [25-28]. One of the most widely applied framework for understanding IS adoption in an organizational context (e.g., [21, 29]) was developed by [30]. The "Technology- Organization- Environment" (TEO) framework concerns contexts surrounding adoption decisions in three main aspects, namely technological context, organizational context and external environment. However, regarding the diffusion of innovations theory [31], adoption decisions typically involve individuals who decide whether to adopt or reject an innovation. Therefore, in addition to the three elements of the TOE framework, this paper also includes an individual context to frame the review of the literature on OSS adoption.

In terms of the "technological context", one of the most suggested factors is the relative advantage of OSS compared to proprietary software in terms of cost and reliability [8, 20, 21, 32]. For instance, the savings for the first phase of OSS implementation in a large UK Hospital were estimated to be around 13 million EUR over five years [32]. Further, it has also been found that the low-cost nature of OSS is likely to attract firms in less-developed countries [2]. Many reliability claims of OSS products seem to go in both directions. Some scholars claim that due to the "bazaar style" development of OSS, its quality and reliability are probably superior to proprietary software [3, 9, 14]. In addition, OSS is recognized as mature particularly in horizontal infrastructure such as operating systems (e.g., Linux) and web or mail servers (e.g., Apache or Sendmail) [7]. By contrast, Dedrick and West [21] argued that, from a user's perspective, OSS in mission critical application domains still lacks reliability, while it is considered reliable enough for

non-critical application domains. However, making comparisons between OSS and proprietary software in general is pointless, because both cover a wide range of software[8]. Consistently, Dedrick and West [20] found that there is little support for the idea that OSS is necessarily more or less reliable than its proprietary counterparts.

Prior works have also indicated that adoption decisions of OSS products are substantially influenced by the issues of compatibility with current systems, skills and requirements [5, 8, 9, 21, 29, 33]. For instance, the most prominent reason why managers rejects OSS in Australia's top firms is that they perceives no relevance to their business in OSS offerings [9].

OSS proponents often identify vendor lock-in avoidance as an important factor. Firms adopt OSS in order to become less dependent on software vendors [8, 20]. An organization that is locked-in to a software vendor might find itself depending on that particular vendor for updates, bug fixes, and services [20]. Nevertheless, the situation where switching costs were high could prevent firms to switch from current proprietary systems to OSS solutions [9].

Regarding the availability of source code, much literature argues that free access to the software source code is not really a factor in the decision to adopt OSS solutions [8, 20, 21, 32]. There are two explanations for this. Firstly, appropriate user experience and skills are required to exploit the flexibility of OSS. Secondly, for mature OSS products such as Linux, Apache or Sendmail, source code modifications may not be necessary [20].

For "organizational context", there are several factors including the adopting firm's IT innovativeness, presence of boundary spanners, top management support and resources. Scholars suggested that a firm's innovation orientation is an important factor in terms of adoption decisions and timing of adoption [2, 21, 34].

Another important driver of OSS adoption is the presence of boundary spanners or in-house OSS champions [29, 35]. The introduction of OSS in organizations is typically a bottom-up initiative in which individuals in the organizations suggest the use of OSS [33, 35]. Those individuals are called "boundary spanners" [35]. These boundary spanners require less training, and are able to encourage and assist their colleagues to get familiar with the OSS solution. Moreover, since in-house expertise is available, drawbacks from the limited external support of OSS could be mitigated [35].

Top management support is crucially important for a venture into an OSS adoption [19, 32]. This is particularly the case for adopting OSS in the domain of highly visible applications [29]. In contrast, conservative nature of top executives is a major barrier to the adoption of OSS [34].

There are two different dimensions (financial versus human resources) that lead to different impacts on OSS adoption in firms [21]. On the one hand, given the limited financial resources, OSS is preferable to proprietary software as the former generally does not require license fee [3, 8]. On the other hand, lack of required technological expertise could hinder OSS adoption [9, 33, 34]. To give an example, reasons why Beaumont hospital decided

to adopt several OSS solutions were the limited IT budget and the presence of OSS-familiar staff [32].

Regarding "environment" external to the adopting firms, several factors appear to influence OSS adoption decisions. They are risk-averse culture, availability of external support, external advices or successful exemplars, and IT-based networks.

Scholars have suggested that general attitude to risk and uncertainty affects decisions to adopt OSS [2, 29]. For example, in a cross-national study, Qu et al. [2] argued that due to the higher reliability and superior security of OSS (compared to proprietary software), firms in countries with higher uncertainty avoidance culture are more likely to adopt OSS.

The next factor is the availability of external support. Although supports from OSS communities via email lists or online forums are generally available [7]; managers still prefer a responsible third party [34]. This leads to a major barrier to OSS adoption, i.e., lack of external support services [7, 9, 20, 33, 34]. However, it should be noted that, in principle, anyone can offer support services for OSS products. Therefore, it is likely that small service-centric IT firms could thrive by providing training, support, and consultancy to local firms that adopt OSS products [7]. Similarly, lack of government support was found to negatively impact OSS adoption in SMEs [19].

Successful exemplars could facilitate OSS adoption. It was found that the non-existence of high-profile successful cases of OSS adoption inhibits OSS adoption in commercial firms [29].

It is suggested that the diffusion of OSS depends on both the initial distribution of beliefs and on the presence of network externality [2, 15, 29]. In certain sectors where interoperability is very important, particular proprietary software may appear to offer a communication standard between firms. Consequently, the possibility of the adoption of OSS would be diminished [29]. On the contrary, it is argued that OSS can flourish even with the presence of proprietary networks because the presence of any IT-based networks would raise awareness of OSS and the open nature of OSS products [2, 15].

Regarding the "individual" context, personal innovativeness in IT and personal perception of OSS maturity are suggested as important OSS adoption facilitators [6, 9, 29, 36]. Individuals who possess a high level of innovativeness in technology are more likely to tolerate the risks of using OSS without the presence of external support services [6].

Many managers believe that software which is available for free is probably not as good as software which is paid for [5, 8]. Goode's [9] study also supports these beliefs. He found that managers in Australia's top firms who rejected OSS thought that OSS was not commercial enough.

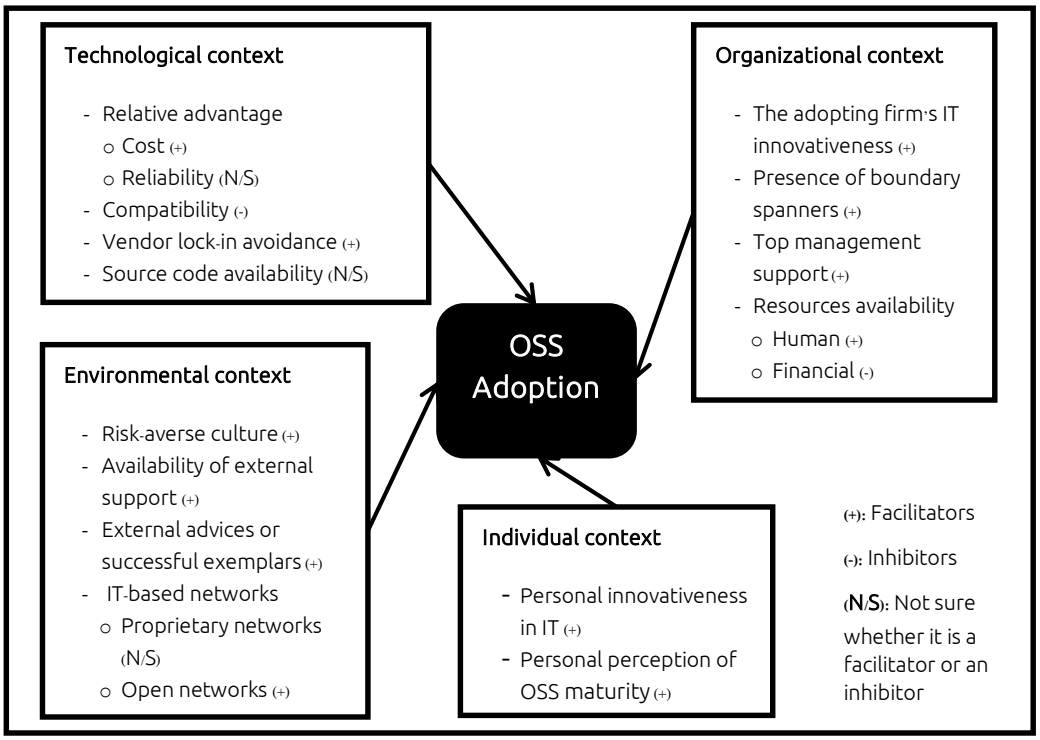

Figure 1. Facilitators and inhibitors of OSS adoption.

Figure 1 summarizes all of the facilitators and inhibitors presented in this subsection.

4. Discussion

From the review, I found that some of the areas are still under-researched and more empirical evidences are needed. Those areas are, for example, how firms build their business models around OSS products [12], and how firms should balance the control of the OSS projects that they have initiated [11]. In addition, the effects of some of the adoption factors presented in the study are still inconclusive, such as OSS reliability, source code availability, and IT-based proprietary networks. A more systematic review or a meta-analysis study focusing on understanding how these factors influence OSS adoption decisions may be fruitful.

In this review, a number of prior research studies (both empirical and review studies) have been reviewed. One limitation of this study is that there is a risk that some articles have been missed in the review process. However, I believe that even if more studies were to be included, it is not unlikely that the major findings would be similar. Furthermore, the author does not claim that the list of facilitators and inhibitors suggested in Figure 1 are exhaustive. Despite that, I argue that these factors are frequently cited in previous studies. Thus, they might be key factors that managers thinking about adopting OSS products should consider. Also, the framework of four core adoption

contexts is proposed to be useful when one makes OSS adoption decisions because different organizations perceive OSS differently. Therefore, the adoption must me established in the context of individual organization [10, 19].

5. Conclusion

This paper reviewed prior studies in two different research streams, namely OSS adoption in software- intensive firms and non- software intensive firms. Research in the first stream explores how software-intensive firms employ OSS products and processes. At least 7 practices have been suggested in the literature. The other research stream focuses on identifying factors that facilitate or inhibit OSS adoption in non-software intensive organizations. A framework containing four core elements – i. e. technological, organizational, external environment, and individual contexts – was employed in order to frame our review. A total of 14 factors were found to impact OSS adoption in commercial firms.

Many of the articles included in this review are in forms of surveys and reviews of the literature, which give broad and essential information for managers who are thinking of adopting OSS products or practices in their organization. However, from the review, I found that more studies investigating specific cases of OSS implementation could possibly provide more knowledge regarding how different firms adopt OSS in their unique contexts. Since one of the most cited factors affecting OSS adoption is costs, an interesting avenue for future research could be to study long-term costs and consequences of using and keeping OSS products operational. Furthermore, research could investigate how software- intensive firms can transform their proprietary software to OSS and build a community around it. Alternatively, another interesting topic could be how non- software- intensive companies can benefit from the concept of openness by collaborating with a community of their lead users to develop new products or services.

REFERENCES

[1] E. Glynn et al., "Commercial adoption of open source software: an empirical study," in International Symposium on Empirical Software Engineering, Noosa Heads, Australia, p. 10,2005.

[2] W. G. Qu et al., "Multi-level framework of open source software adoption," Journal of Business Research, vol. 64, pp. 997-1003, 2011.

[3] S. Krishnamurthy, "A managerial overview of open source software", Business Horizons, vol. 46, pp. 47-56, 2003.

[4] A. Mockus et al., "Two case studies of open source software development: Apache and Mozilla", ACM Trans. Softw. Eng. Methodol., vol. 11, pp. 309-346, 2002.

[5] D. Nagy et al., "Organizational adoption of open source software: barriers and remedies", Commun. ACM, vol. 53, pp. 148-151, 2010.

[6] K. L. Gwebu and J. Wang, "Adoption of Open Source Software: The role of social identification" , Decision Support Systems, vol. 51, pp. 220-229, 2011.

[7] B. Fitzgerald, "The Transformation of Open Source Software," MIS Quarterly, vol. 30, pp. 587-598, 2006.

[8] K. Ven et al., "Should You Adopt Open Source Software?" , Software, IEEE, vol. 25, pp. 54-59, 2008.

[9] S. Goode, " Something for nothing: management rejection of open source software in Australia's top firms", Information & Management, vol. 42, pp. 669-681, 2005.

[10] S. Ziemer et al., "Understanding Open Source in an Industrial Context" , in Signal Image Technology and Internet Based Systems, SITIS '08. IEEE International Conference on, pp. 539-546, 2008.

[11] Ø. Hauge, C. Ayala, and R. Conradi, "Adoption of open source software in software-intensive organizations – A systematic literature review", Information and Software Technology, vol. 52, pp. 1133-1154, 2010.

[12] M. Höst and A. Oručević-Alagić, "A systematic review of research on open source software in commercial software product development", Information and Software Technology, vol. 53, pp. 616-624, 2011.

[13] G. Hertel et al. , " Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel" , Research Policy, vol. 32, pp. 1159-1177, 2003.

[14] E. S. Raymond, "The cathedral and the bazaar : musings on Linux and Open Source by an accidental revolutionary" , Rev. ed. Beijing ; Cambridge, Mass.: O'Reilly, 2001.

[15] A. Bonaccorsi and C. Rossi, "Why Open Source software can succeed," Research Policy, vol. 32, pp. 1243-1258, 2003.

[16] Android. (2015, 5 Oct) . Android Open Source Project License. Available: https://source.android.com/source/licenses.html#android-open-source-project-license

[17] K. L. Gwebu and J. Wang, "Seeing eye to eye? An exploratory study of free open source software users' perceptions" , Journal of Systems and Software, vol. 83, pp. 2287-2296, 2010.

[18] S. Y. Sohn and M. S. Mok, "A strategic analysis for successful open source software utilization based on a structural equation model" , Journal of Systems and Software, vol. 81, pp. 1014-1024, 2008.

[19] R. D Macredie and K. Mijinyawa, "A theory-grounded framework of Open Source Software adoption in SMEs," Eur J Inf Syst, vol. 20, pp. 237-250, 2011.

[20] J. Dedrick and J. West, "Movement ideology vs. user pragmatism in the organizational adoption of open source software" , in Computerization Movements and Technology Diffusion, From Mainframes to Ubiquitous Computing, K. L. Kraemer and M. Elliott, Eds., ed Melford, NJ: Information Today, 2007.

[21] J. Dedrick and J. West, "An exploratory study into open source platform adoption" , in System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, 2004.

[22] J. Feller and B. Fitzgerald, "Understanding Open Source Software development" , London ; Boston: Addison-Wesley, 2002.

[23] Ø. Hauge et al., "Surveying Industrial Roles in Open Source Software Development", in Open Source Development, Adoption and Innovation. vol. 234, J. Feller, B. Fitzgerald, W. Scacchi, and A. Sillitti, Eds., ed: Springer US, pp. 259-264, 2007.

[24] J. West, "How open is open enough?: Melding proprietary and open source platform strategies", Research Policy, vol. 32, pp. 1259-1285, 2003.

[25] L. G. Tornatzky et al., "The processes of technological innovation" , Lexington, Mass.: Lexington Books, 1990.

[26] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology" , MIS Quarterly, vol. 13, pp. 319-340, 1989.

[27] J. K. Liker and A. A. Sindi, "User acceptance of expert systems: a test of the theory of reasoned action", Journal of Engineering and Technology Management, vol. 14, pp. 147-173, 1997.

[28] S. Taylor and P. Todd, "Decomposition and crossover effects in the theory of planned behavior: A study of consumer adoption intentions", International Journal of Research in Marketing, vol. 12, pp. 137-155, 1995.

[29] E. Glynn et al., "Commercial adoption of open source software: an empirical study" , International Symposium on Empirical Software Engineering, Noosa Heads, Australia, 2005.

[30] R. Depietro et al., "The context for change: Organization, technology and environment", in The processes of technological innovation, L. G. Tornatzky and M. Fleischer, Eds., ed Lexington, Mass.: Lexington Books, pp. 151-175, 1990

[31] E. M. Rogers, Diffusion of innovations, 4th ed. New York: Free Press, 1995.

[32] B. Fitzgerald and T. Kenny, "Open source software the trenches: Lessons from a large-scale OSS implementation," in ICIS Proceedings, pp. 27, 2003.

[33] L. Morgan and P. Finnegan, "How Perceptions of Open Source Software Influence Adoption: An Exploratory Study," in 15th European Conference on Information Systems, St. Gallen, Switzerland, pp. 973-984, 2007

[34] G. Paré et al., "Barriers to open source software adoption in Quebec' s health care organizations," Journal of medical systems, vol. 33, pp. 1-7, 2009.

[35] K. Ven and J. Verelst, " The Organizational Adoption of Open Source Server Software: A Quantitative Study," in ECIS, Galway, Ireland, pp. 1430-1441, 2008.

[36] Y. Li et al., "Open source software adoption: motivations of adopters and amotivations of non-adopters," The DATA BASE for Advances in Information Systems, vol. 42, pp. 76-94, 2011.