International Journal of u- and e- Service, Science and Technology
Vol.9, No. 6 (2016), pp.115-124
http://dx.doi.org/10.14257/ijunesst.2016.9.6.11

The Impact of Community and Commercial Open Source Software on the Quality Strategies of Software Producers

Mingqing Xing[1] and Lu Shen[2]

[1]*School of Economics and Management, and Neural Decision Science Laboratory, Weifang University, Weifang, Shandong, 261061, China*
[2]*School of Economics and Business Administration, Central China Normal University, Wuhan, Hubei, 430079, China*
[1]*mqxing1979@163.com*

Abstract

This study investigates how open source software can play influences on the quality choices of commercial open source and proprietary software providers. It considers two types of open source software: community open source and commercial open source, and assumes that the usability of commercial open source software is better than community open source substitute but inferior to proprietary substitute. It finds that: (i) the functional quality of proprietary software decreases as the functional quality of community open source software increases, but it may increase as the usability of community open source software increases; (ii) even if commercial open source producers must open the source codes of their quality contributions, they have incentive to enhance the functional quality of community open source software; (iii) the influence of community open source software's quality (usability or functional quality) or commercial open source software's usability on the functional quality of commercial open source and proprietary software may be not same; (iv) the appearance of commercial open source software may lead to proprietary software producers lowering their software functional quality.

Keywords*: software quality, community open source software, commercial open source software, proprietary software, software function, software usability*

1. Introduction

In recent years, open source software has become more prominent in the software industry. As opposed to proprietary software, open source software allows the users to share its source codes, identify and correct errors, and redistribute the software to other users [1]. Linux (an operating system) and Apache (a web server) are typical examples of open source software. Open source software has become an increasingly threatening competitor to traditional proprietary software. A few researchers have examined the competition between these two types of software. For examples, Dalle and Jullien [2] and Xing [3] investigated the technological and quantity competition between open source and proprietary software providers respectively; Lin [4] and Gramstad [5] studied the impact of user skills, network effects and piracy on the competition between proprietary and open source software respectively. The above studies only take into account community (or free) open source software but do not consider commercial open source software.

Open source software can be divided into two categories: community open source and commercial open source [6]. Community open source software is owned by a community whose members don't obtain direct revenues from the software, while commercial open source software is owned by a software provider whose goals is to gain profit from the software. In the study of Kumar *et al.* [7], commercial open source software is defined as the privately built software product based on the publicly available source codes. Recent years more software producers develop commercial products on the basis of open source

ISSN: 2005-4246 IJUNESST

software. A typical example is Red Hat Inc, whose commercial versions of the free available Linux are designed to enhance the usability and features of the Linux operating system. There are some studies involving commercial open source software. For examples, Riehle [6] and Widenius and Nyman [8] characterized the commercial use of open source and gave the commercial open source business models; Kumar *et al.* [7] considered the product innovations for commercial open source vendors; Sen [1] and Xing [9] focused on the price competition between proprietary software and its commercial open source alternatives. Although the above findings relate to commercial open source software, they do not study its influence on the quality choices for proprietary software producers. The producers of proprietary software face many challenges in today's competitive market environment. An important challenge comes from commercial open source software. Due to the property of owners for commercial and community open source software is different, this may lead to diverse strategic reactions for proprietary producers when they compete with open source software. In addition, commercial open source providers also need to know their quality choices in face of competition from community open source and proprietary software. By considering both community and commercial open source, this study examines the impact of open source software on the quality choices for commercial open source and proprietary software producers.

The remainder of this study is organized as follows. The basic model is given in Section 2. Section 3 considers the case that proprietary software competes with community open source substitute. Section 4 analyzes the case that proprietary software competes with both community and commercial open source substitutes. The equilibrium results in Section 3 and Section 4 are compared in Section 5. The final section concludes this paper.

2. The Model

This paper studies how open source software affects the quality choices of software producers. Section 3 considers the case of proprietary software only competing with community open source substitute in a software market (Case A). The case of proprietary software facing competition from both community and commercial open source software is investigated in Section 4 (Case B). Section 5 compares the equilibrium results in above two cases. This study makes a fundamental distinction between community and commercial open source software [6]. The former is open source software owned by a not for-profit community, while the latter is open source software owned by a commercial software producer. Commercial open source vendors build software products based on community open source software with the purpose of deriving revenues from the software.

Community open source software lacks usability and is generally less user-friendly in contrast to its proprietary software substitute. Therefore, the users need to have significant expertise to adopt community open source software effectively [10]. The quality of a software product depends on its usability (measured by the ease of installation, user interface, documentation and the level of technical support) and functional quality (measured by feature set, security, reliability, *etc*) [11-12]. Commercial open source producers add value to community open source software by increasing the functionality and/or usability [7].

Recent surveys show that the technical skills of users are one of the key factors in their open source adoption decisions [4]. This study models users in the software market as heterogeneous in their level of technical skills, which is measured by parameter θ. As the treatment of Choudhary and Zhou [12], the level of users' technical skills decreases with θ. In general, the highly skilled users may not place as much emphasis on the software usability as the less skilled users when they deploy a software product [1]. This study assumes that a user's willingness to pay for the software usability decreases with his/her

level of technical skills. The total number of users is normalized to 1 and they are uniformly distributed over the unit interval $[0,1]$. Note that the parameter θ is closed to 0 for a highly skilled user and closed to 1 for a lowly skilled user.

The indirect utility function for the generic user at $\theta \in [0,1]$ when he/she adopts software i is defined as:

$$S_i = \theta u_i + f_i - p_i,\ i = o, c, p \tag{1}$$

The subscripts 'o', 'c' and 'p' denote community open source, commercial open source and proprietary software respectively. In (1), u_i, f_i and p_i denote the usability, functional quality and price of software i respectively. Note that: (i) the price of community open source software equals zero because it can be freely available from open source community (i.e., $p_o = 0$); (ii) commercial open source software is assumed to be better than community open source software but inferior to proprietary software in the software usability (*i.e.*, $u_o < u_c < u_p$).

This paper mainly examines the functional quality decisions for commercial open source and proprietary producers. For convenience of analysis, the functional quality and usability of community open source software and the usability of commercial open source and proprietary software are assumed to be exogenously determined. Moreover, we assume that the model parameters can ensure both community open source software and proprietary software (resp. community open source software, commercial open source software and proprietary software) present in the software market in Case A (resp. Case B) and the profit for both commercial open source and proprietary producers is not negative.

3. Case A: Proprietary Software Competes with Community Open Source Software

This section considers the case of proprietary software only facing competition from community open source substitute. The marginal user who is indifferent between deploying proprietary and community open source software, indexed by $\bar{\theta}$, is given by $S_o = S_p$:

$$\bar{\theta} u_o + f_o = \bar{\theta} u_p + f_p - p_p \tag{2}$$

Solving (2) gives:

$$\bar{\theta} = \frac{f_o - f_p + p_p}{u_p - u_o} \tag{3}$$

The software market is assumed to be fully covered. Software users with a type $\theta < \bar{\theta}$ will adopt community open source software and software users with a type $\theta > \bar{\theta}$ will prefer proprietary software. It implies that the demand functions for open source community and proprietary producer are:

$$D_o = \int_0^{\bar{\theta}} 1\, d\theta = \bar{\theta} = \frac{f_o - f_p + p_p}{u_{po}} \tag{4}$$

$$D_p = \int_{\bar{\theta}}^{1} 1\, d\theta = 1 - \bar{\theta} = 1 - \frac{f_o - f_p + p_p}{u_{po}} \tag{5}$$

In (4) and (5), $u_{po} = u_p - u_o$. The resulting profit function of proprietary producer is:

$$\Pi_p = \frac{p_p[u_{po} - (f_o - f_p) - p_p]}{u_{po}} - rf_p^2 \tag{6}$$

In (6), rf_p^2 is cost that proprietary producer invests in quality, where r is a scaling parameter for this cost.

The timing of the model in this section is as follows. Software producer chooses its software functional quality firstly and then sets the price of its software.

Proprietary producer pursues profit maximization. The first order condition yields the equilibrium price of proprietary software:

$$p_p^* = \frac{u_{po} - (f_o - f_p)}{2} \tag{7}$$

The resulting profit function for proprietary producer on variable f_p is:

$$\Pi_p = \frac{[u_{po} - (f_o - f_p)]^2}{4u_{po}} - rf_p^2 \tag{8}$$

Solving the first order condition gives the equilibrium quality of proprietary software:

$$f_p^* = \frac{u_{po} - f_o}{4ru_{po} - 1} \tag{9}$$

The second order condition requires that the model parameters meet: $r > \frac{1}{4u_{po}}$. In order to guarantee the positive demands for both community open source software and proprietary software, the following inequality must hold: $\frac{1}{2r} - u_{po} < f_o < u_{po}$.

Proposition 1. (i) f_p^* decreases with f_o; (ii) when $f_o < \frac{1}{4r}$, f_p^* increases with u_o; when $f_o > \frac{1}{4r}$, f_p^* decreases with u_o.

Proof. (i) According to (9), $\frac{\partial f_p^*}{\partial f_o} = -\frac{1}{4ru_{po} - 1} < 0$; (ii) According to (9), $\frac{\partial f_p^*}{\partial u_o} = \frac{1 - 4rf_o}{(4ru_{po} - 1)^2}$. Thus, $\frac{\partial f_p^*}{\partial u_o} > 0$ if $f_o < \frac{1}{4r}$, and $\frac{\partial f_p^*}{\partial u_o} < 0$ if $f_o > \frac{1}{4r}$.

The above proposition shows that, the functional quality of proprietary software decreases as the functional quality of community open source software increases. The reason is that, when the functional quality of community open source software increases, the price (resp. demand) of proprietary software lowers (resp. decreases). Consequently the profit of proprietary producer decreases. This leads to a decrease of the funds of proprietary producer for raising product quality. However, the functional quality of proprietary software may increase with the usability of community open source software. The reason is that, the increase of community open source software's usability makes two effects on proprietary producer. On the one hand, it leads to a decline in the price of proprietary software. On the other hand, it leads to an increase of the demand of proprietary software if the functional quality of community open source software is low enough ($f_o < f_p$) and the opposite appears if the functional quality of community open source software is high enough ($f_o > f_p$). When the functional quality of community

open source software is sufficient low and the latter effect is greater than the former one, the increase of community open source software's usability leads to an increase of the proprietary producer's profit. Consequently the proprietary producer has more money to improve its software functionality.

4. Case B: Proprietary Software Competes with Both Community and Commercial Open Source Software

This section analyzes the case that commercial open source software, provided by another software producer, also appears in the software market. Thus, proprietary software competes with both community and commercial open source substitutes. When commercial open source vendors develop software products on the basis of community open source software, they must follow its open source licenses. The General Public License (GPL) is the most common open source license, which requires software developers to open their development of features. Software developers and open source community can wholly derive each other's feature contributions under this license.

This study only analyzes the case that commercial open source producer builds commercial software products based on community open source software under the GPL, and assumes that open source community completely adopts the feature contributions of commercial open source producer (*i.e.*, f). Under this condition, the functional quality of both community and commercial open source software equals f_c (*i.e.*, $f_c = f_o + f$) if commercial open source producer improves the functional quality of community open source software. Thus, the indirect utility functions for user with θ when he/she deploys community and commercial open source software are respectively given by: $S_o = \theta u_o + (f_o + f)$ and $S_c = \theta u_c + (f_o + f) - p_c$. The marginal user who is indifferent between adopting community and commercial open source software, indexed by $\hat{\theta}$, is given by $S_o = S_c$:

$$\hat{\theta} u_o + (f_o + f) = \hat{\theta} u_c + (f_o + f) - p_c \tag{10}$$

Solving (10) gives:

$$\hat{\theta} = \frac{p_c}{u_c - u_o} \tag{11}$$

The marginal user who is indifferent between adopting commercial open source and proprietary software (indexed by $\check{\theta}$) is given by $S_c = S_p$:

$$\check{\theta} u_c + (f_o + f) - p_c = \check{\theta} u_p + f_p - p_p \tag{12}$$

Solving (12) yields:

$$\check{\theta} = \frac{f_o + f - f_p + p_p - p_c}{u_p - u_c} \tag{13}$$

We suppose that $\hat{\theta}$ and $\check{\theta}$ satisfy the following inequality: $0 < \hat{\theta} < \check{\theta} < 1$. This assumption ensures that the demands of three types of software are positive and they are all less than 1. According to (11) and (13), the demand functions for open source community, commercial open source producer and proprietary producer are:

$$D_o = \int_0^{\hat{\theta}} 1 \, d\theta = \hat{\theta} = \frac{p_c}{u_{co}} \tag{14}$$

$$D_c = \int_{\theta}^{\hat{\theta}} 1 \, d\theta = \hat{\theta} - \theta = \frac{f_o + f - f_p + p_p - p_c}{u_{pc}} - \frac{p_c}{u_{co}} \tag{15}$$

$$D_p = \int_{\hat{\theta}}^{1} 1 \, d\theta = 1 - \hat{\theta} = 1 - \frac{f_o + f - f_p + p_p - p_c}{u_{pc}} \tag{16}$$

In (14), (15) and (16), $u_{co} = u_c - u_o$ and $u_{pc} = u_p - u_c$. Thus, the profit functions for three types of software providers are:

$$\Pi_o = 0 \tag{17}$$

$$\Pi_c = p_c D_c - rf^2 = p_c \left(\frac{f_o + f - f_p + p_p - p_c}{u_{pc}} - \frac{p_c}{u_{co}} \right) - rf^2 \tag{18}$$

$$\Pi_p = p_p D_p - rf_p^{\,2} = p_p \left(1 - \frac{f_o + f - f_p + p_p - p_c}{u_{pc}} \right) - rf_p^{\,2} \tag{19}$$

In this section, the timing of the model is as follows. Both commercial open source and proprietary producers decide their software functional quality in the first stage and they price their software in the second stage.

Assume that both proprietary and commercial open source producers seek the maximum of profit. The first order conditions yield the equilibrium prices:

$$p_c^{*'} = \frac{u_{co}(u_{pc} + f_o + f - f_p)}{4u_{po} - u_{co}} \tag{20}$$

$$p_p^{*'} = \frac{2u_{po}u_{pc} - (2u_{po} - u_{co})(f_o + f - f_p)}{4u_{po} - u_{co}} \tag{21}$$

The resulting profit functions on variables f and f_p are given by:

$$\Pi_c = \frac{u_{co}u_{po}(u_{pc} + f_o + f - f_p)^2}{u_{pc}(4u_{po} - u_{co})^2} - rf^2 \tag{22}$$

$$\Pi_p = \frac{[2u_{po}u_{pc} - (2u_{po} - u_{co})(f_o + f - f_p)]^2}{u_{pc}(4u_{po} - u_{co})^2} - rf_p^{\,2} \tag{23}$$

The first order conditions give the equilibrium results:

$$f^{*'} = \frac{u_{co}u_{po}[r(4u_{po} - u_{co})(u_{pc} + f_o) - (2u_{po} - u_{co})]}{r(4u_{po} - u_{co})[ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}]} \tag{24}$$

$$f_p^{*'} = \frac{(2u_{po} - u_{co})[r(4u_{po} - u_{co})[2u_{po}u_{pc} - (2u_{po} - u_{co})f_o] - u_{co}u_{po}]}{r(4u_{po} - u_{co})[ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}]} \tag{25}$$

Consequently the equilibrium functional quality of commercial open source software equals:

$$f_c^{*'} = f^{*'} + f_o = \frac{u_{co}u_{po}[r(4u_{po} - u_{co})(u_{pc} + f_o) - (2u_{po} - u_{co})]}{r(4u_{po} - u_{co})[ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}]} + f_o \tag{26}$$

The second order conditions require that the model parameters meet: $r > \frac{(2u_{po} - u_{co})^2}{u_{pc}(4u_{po} - u_{co})^2}$. Moreover, in order to make sure that the demands for all software providers are positive and the total demand equals 1, the following inequality must hold:

$$\frac{2u_{po} - u_{co}}{r(4u_{po} - u_{co})} - u_{pc} < f_o < \frac{2ru_{po}u_{pc}(4u_{po} - u_{co}) - u_{co}u_{po}}{r(4u_{po} - u_{co})(2u_{po} - u_{co})} \quad (27)$$

Proposition 2. $f^{*'} > 0$.

Proof. Because the demands for all software providers and the functional quality for proprietary software are positive in equilibrium, the following inequalities hold: $r(4u_{po} - u_{co})(u_{pc} + f_o) - (2u_{po} - u_{co}) > 0$ and $r(4u_{po} - u_{co})[ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}] > 0$. Combing with (24), we obtain $f^{*'} > 0$.

Proposition 2 indicates that, commercial open source producers have incentive to improve the functional quality of community open source software even if they must open the software source codes of its quality contributions and open source community can completely absorb these contributions. The reason is that, although the increase of commercial open source software's functional quality may not promote the competitive advantage of commercial open source software relative to community open source software, it can improve the profitability of commercial open source producer in face of competition from proprietary software. This may explain why Red Hat Inc (a commercial open source provider) makes great contributions to Linux Kernel, Linux X Windows System and others, despite these contributions must make publicly available under the General Public License [7].

Proposition 3. $f^{*'}$ and $f_c^{*'}$ increase with f_o, while $f_p^{*'}$ decreases with f_o.

Proof. According to (24), (25) and (26), $\frac{\partial f^{*'}}{\partial f_o} = \frac{u_{co}u_{po}}{ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}} > 0$, $\frac{\partial f_c^{*'}}{\partial f_o} = \frac{u_{co}u_{po}}{ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}} + 1 > 0$ and $\frac{\partial f_p^{*'}}{\partial f_o} = -\frac{(2u_{po} - u_{co})^2}{ru_{pc}(4u_{po} - u_{co})^2 - (2u_{po} - u_{co})^2 - u_{co}u_{po}} < 0$.

The above proposition shows that, the impact of the functional quality of community open source software on the functional quality of commercial open source software and proprietary software is different. The former (resp. the latter) increases (resp. decreases) as the functional quality of community open source software increases. This is because the increase of the functional quality of community open source software improves the competitive advantage for both community and commercial open source software relative to proprietary software [18]. Consequently the commercial open source producer (resp. proprietary producer) has higher (resp. lower) incentive to improve the functional quality of its software.

By means of numerical analysis, we investigate how the usability of community open source software affects the equilibrium results (see Table 1 and Table 2).

Table 1. Effect of u_o on $f^{*'}$, $f_c^{*'}$ and $f_p^{*'}$ when $f_o = 0.1$ ($u_c = 1$, $u_p = 2$ and $r = 2$)

u_o	$f^{*'}$	$f_c^{*'}$	$f_p^{*'}$
0.2	0.0175	0.1175	0.1236
0.4	0.0141	0.1141	0.1247
0.6	0.0102	0.1102	0.1259
0.8	0.0055	0.1055	0.1272

Table 2. Effect of u_o on $f^{*'}$, $f_c^{*'}$ and $f_p^{*'}$ When $f_o = 0.3$ ($u_c = 1$, $u_p = 2$ and $r = 2$)

u_o	$f^{*'}$	$f_c^{*'}$	$f_p^{*'}$
0.2000	0.0214	0.3214	0.1021
0.4000	0.0173	0.3173	0.1020
0.6000	0.0125	0.3125	0.1018
0.8000	0.0068	0.3068	0.1013

According to Table 1 and Table 2, the following results can be found.

Proposition 4. (i) $f^{*'}$ and $f_c^{*'}$ decrease with u_o; (ii) $f_p^{*'}$ increases with u_o if f_o is small, while it decreases with u_o if f_o is big.

The first part of Proposition 4 implies that, the functional quality of commercial open source software decreases as the usability of community open source software increases. The reason is that, when the usability of community open source software increases, it is closer to the usability of commercial open source software. This leads to more intense competition between community and commercial open source software. Therefore, the profitability of commercial open source producer declines and the commercial open source producer has less incentive to improve the functional quality of its software. The second part of Proposition 4 demonstrates that, the functional quality of proprietary software increases as the usability of community open source software increases if the functional quality of community open source software is small enough, while the opposite appears if the functional quality of community open source software is big enough. Note that the impact of the usability of community open source software on the functional quality of commercial open source software and proprietary software may be different.

Now we analyze the impact of the usability of commercial open source software on the functional quality of commercial open source software and proprietary software. Numerical examples are shown in the following table.

Table 3. Effect of u_c on $f^{*'}$, $f_c^{*'}$ and $f_p^{*'}$ ($f_o = 0.1$, $u_o = 0.5$, $u_p = 1.5$ and $r = 2$)

u_c	$f^{*'}$	$f_c^{*'}$	$f_p^{*'}$
0.7000	0.0067	0.1067	0.1276
0.9000	0.0152	0.1152	0.1251
1.1000	0.0264	0.1264	0.1197
1.3000	0.0485	0.1485	0.1002

According to Table 3, the following conclusion can be found.

Proposition 5. $f^{*'}$ and $f_c^{*'}$ increase with u_c, while $f_p^{*'}$ decreases with u_c.

Proposition 5 shows that, when the usability of commercial open source software increases, the commercial open source producer will improve the functional quality of software, while the proprietary producer will lower the functional quality of software. The intuition behind this result is that, the increase of the usability of commercial open source software leads to the competitive advantage increase (resp. decrease) for commercial open source software (resp. proprietary software) in face of competition from proprietary software (resp. commercial open source software). Thus, the commercial open source producer (resp. proprietary producer) has more (resp. less) money to improve its software functional quality. Comparing Proposition 5 and Proposition 4, we find that the impact of commercial and community open source software's usability on the quality choices of commercial open source and proprietary producers may be different.

5. Comparisons

This section compares the equilibrium functional qualities of proprietary software in Case A and Case B and then examines how the appearance of commercial open source software affects the functional quality of proprietary software. According to (9), $f_p^* = 0.3$ if $f_o = 0.1$, $u_o = 0.5$, $u_p = 1.5$ and $r = 2$. Combing with the values of $f_p^{*'}$ in Table 3, we can obtain the following conclusion.

Proposition 6. $f_p^* > f_p^{*'}$.

The above proposition indicates that, in comparison with the case that proprietary software only competes with community open source software, the presence of commercial open source software leads to the functional quality of proprietary software decrease. This is because, compared with community open source software, the usability of its commercial open source substitute is better. The presence of commercial open source software intensifies the competition between open source software and proprietary software. In contrast to the case of proprietary software only competing with community open source software, the proprietary producer obtains less profit when commercial open source software also appears in the market. Consequently it has lower incentive to improve its software functional quality under Case B than Case A.

6. Conclusions

By constructing mathematical models, this study examines the impact of competition from community and commercial open source software on the strategic quality for software producers. Assume that the usability of commercial open source software is inferior to proprietary software and better than community open source software. This study considers two cases: (i) proprietary software only competes with community open source software (Case A); (ii) proprietary software competes with both community and commercial open source software (Case B). It mainly finds that: (i) the functional quality of proprietary software decreases with the functional quality of community open source software, while it may increase with the usability of community open source software under both Case A and Case B; (ii) commercial open source producers have incentive to improve the functional quality of community open source software even if they must open the software source codes of their quality contributions and the open source community can completely adopt these contributions under Case B; (iii) the impact of community open source software's usability and functional quality and commercial open source software's usability on the functional quality of commercial open source software and proprietary software may be different under Case B; (iv) compared Case A and Case B,

the presence of commercial open source software leads to the decrease of the functional quality of proprietary software.

Acknowledgments

The authors would like to thank anonymous referees for their valuable comments. We also acknowledge the financial support from the Natural Science Foundation of Shandong Province (No. ZR2013GL005) and the Beijing Higher Education Young Elite Teacher Project (No. YETP1949).

References

[1] R. Sen, "A Strategic Analysis of Competition between Open Source and Proprietary Software", Journal of Management Information Systems, vol. 24, no. 1, **(2007)**, pp. 238-258.
[2] J. Dalle and N. Jullien, "Open-source vs. Proprietary Software", Working Paper, Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.6144&rep=rep1&type=pdf, **(2002)**.
[3] M. Q. Xing, "The Quantity Competition between Open Source and Proprietary Software", Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering, Yunnan, China **(2010)** November 26-28.
[4] L. H. Lin, "Impact of User Skills and Network Effects on the Competition between Open Source and Proprietary Software", Electronic Commerce Research and Applications, vol. 7, no. 1, **(2008)**, pp. 68-81.
[5] A. R. Gramstad, "Piracy in Commercial vs. Open-source Software Competition", Proceedings of the 41st Annual Conference of the European Association for Research in Industrial Economics, Bocconi, Italy, **(2014)** August 29-31.
[6] D. Riehle, "The Single-vendor Commercial Open Source Business Model", Information Systems and e-Business Management, vol. 10, no. 1, **(2012)**, pp. 5-17.
[7] V. Kumar, B. R. Gordon and K. Srinivasan, "Competitive Strategy for Open Source Software", Marketing Science, vol. 30, no. 6, **(2011)**, pp. 1066-1078.
[8] M. M. Widenius and L. M. Nyman, "The Business of Open Source Software: a Primer", Technology Innovation Management Review, no. 1, **(2014)**, pp. 4-11.
[9] M. Q. Xing, "The Impact of Commercial Open Source Software on Proprietary Software Producers and Social Welfare", Journal of Industrial Engineering and Management, vol. 7, no. 5, **(2014)**, pp. 1183-1196.
[10] K. R. Lakhani and E. V. Hippel, "How Open Source Software Works: 'Free' User-to-user Assistance", Research Policy, vol. 32, no. 6, **(2003)**, pp. 923-943.
[11] H. W. Jung, S. G. Kim and C. S. Chung, "Measuring Software Product Quality: a Survey of ISO/IEC 9126", IEEE Software, vol. 21, no. 5, **(2004)**, pp. 88-92.
[12] V. Choudhary and Z. Z. Zhou, "Impact of Competition from Open Source Software on Proprietary Software", Proceeding of the Institute for Operations Research and the Management Sciences Annual Meeting, Washington, America, **(2007)** November 4-7.

Authors

Mingqing Xing, he is an associate professor in the school of economics and management at Weifang University, China. He received PhD in College of Science from China Agriculture University (2009). His current research interests mainly focus on software industry competition. His research is published in scientific journals such as "Economic Modelling" and "Journal of Industrial Engineering and Management".

