

INCENTIVE SYSTEMS BEHIND OPEN SOURCE SOFTWARE DEVELOPMENT

The paper considers different theories of software development seeking for an answer to the following question: Why do all those numerous user-developers devote a significant part of their time and efforts to produce a virtually free public good? Along with standard economic approach borrowed from the theory of the firm, new theories particular for specifically open source software development are also present.

Introduction

The paper considers incentives affecting developers in the open source (OS thereafter) software. The most vivid examples of OS projects are Linux operating system (in particular, Linux kernel, because there are different distributions of Linux manufactured commercially), KDE and GNOME graphical user interfaces (GUI) (Windows-like shells) Apache web server, StarOffice/OpenOffice office suit, and programming languages Perl and Python. The distinctive feature of the OS is the availability of the source code (a text of a program written in human readable programming language) to a general public, so that any user of such software can evaluate its functionality and even potentially make own improvements. In this respect, the open

source programs should be distinguished from the closed or proprietary source products distributed in a form of compiled binary code. As a matter of fact, it is impossible to obtain the source code from the binary code ("to decompile a program").

The economic organization of software production under OS is drastically different from writing proprietary code by specialized firms. Open source products are created by a large group of independent developers who are dispersed around the world and make comparatively little contributions to the common pool (the "project") via email and discussing lists. These lists are maintained by enthusiasts from within the community, most often by initiators and informal project leaders. Contributions to projects are accepted by voting among the most acknowledged members or by the leaders

alone. The software is available to general public for free or at a very low shipping and handling fee. However, licensing agreements in OS do not prevent manufacturers from charging a positive price. On the other hand, usually individual developers due to the lack of any formal financial organization do not receive any benefits except own satisfaction from their work. Therefore, the major question is "What are the incentives, if not striving for profits, which drive user-developers to spend a significant amount of their time, effort and expertise to contribute to the OS movement?"

First works in the field were merely philosophical and institutional introduction into the world of the open source. The first is "Cathedral and the Bazaar" by Eric Raymond (2001), and another is "Open Source: Voices from the Open Source Revolution" (ed. DiBona, 1999). The article by Brian Behlendorf (1999) from this book is of a particular interest for academic research. Despite the fact that both books do not contain formal economic models or any rigorous data analysis, they both contribute to understanding of the issue and set up the hypothesis to be tested in the successive work. They may serve as a starting point for exploring OS movement.

The paper builds as following. First, I will describe the incentives models common in the literature on the theory of the firm. I will show then, that some of those incentives are applicable to OS movement, whereas others may require certain corrections. Then I will describe hypothesis about incentives pertaining specifically to OS development. Finally, I will present descriptive statistical results conforming or contradicting the theories.

Incentives in the theory of the firm

The literature has paid lots of attention to incentive systems when economic activity is organized in a firm. Holmstrom and Milgrom (1994) summarize approaches undertaken by a scope of authors in this issue. They refer to incentive systems in the realm of make-or-buy decision by a firm and consider a firm as a mechanism using different combinations of incentive instruments to make an employee to perform assigned tasks and to prevent possible shirking on the job. In particular, they consider three major instruments:

- *Pay based on measured performance.* This instrument requires that outcome would be measured individually for every worker, and also claims that significant monitoring costs are involved. Efficient monitoring and work assessment is discussed broadly in the principal-agent literature. One might

refer to works by Alchian and Demsetz (1972), Holmstrom (1982), Milgrom and Roberts (1992).

- *Asset ownership.* This literature is more concentrated on managers' performance and is presented by works of Klein *et al.* (1978) and Williamson (1985). The logic behind this instrument is that if a worker maintains residual claimancy on a firm's assets, he is more likely to work more efficiently.

- *Design of the job.* Coase (1937) and Simon (1951) emphasized a firm's discretion to specify and/or direct worker's activities, work rules, working hours and similar policies that may restrict employee's freedoms.

Baker, Jensen, and Murphy (1988) provide another extensive study of workers incentives within a firm. They attempted to bring together economic theories of monetary compensation, and real - life cases of promotion - based incentive systems, profit-sharing plans, egalitarian pay. They admit that "economic understanding of internal incentive structures is far from complete... It may be that psychologists, behaviorists, human resource consultants, and personal executives understand something about human behavior and motivation that is not yet captured in our economic models" Nevertheless, their contribution is vital to understanding incentives of *employees*. It looks like the major motif found in the incentives literature is immediate or postponed gain of monetary benefits, as if every rational economic agent maximizes present value of current and expected income streams choosing his level of efforts,

Theories about Incentive systems in software development

In his book, Jensen (1998) provides a whole array of possible motivations relevant for every agent, irrespective of organization he is associated with: "The individual cares about almost everything: knowledge, independence, the plight of others, the environment, honor, interpersonal relationships, status, peer approval, group norms, culture, wealth, rules of conduct, and so on... and always allows trade-offs and substitutions."

The major question (What are the incentives?), however, remains intact if the economic activities are not performed under the roof of a firm. Whether theoretical and empirical findings will still be valid? Are there any new theories about to emerge?

a) Proprietary Software Development

The software development is a complex process involving intensive exchange of the knowledge between everybody involved. Usually, the software

is created in independent modules and thus coordination effort is absolutely necessary to make sure that all modules are compatible with each other and gain the necessary result. Moreover, it is not always possible to predict the effect of one module on the work of others. On the other hand, users do not usually care about the exact composition of the software, but rather only about its smooth and bug-free work. Consequently, the price and the value of a product are determined by its functionality as of a whole, not of its constituting parts.

Even though it is possible for a product manager to assess the quality of individual module, it is hard to evaluate a contribution of this module to the work of the whole program. It is always the case with the proprietary software that property rights on the program belong to a firm and are protected by a corresponding license. Every user of such software when buying a copy does not buy a right to own, but rather only a right to install and use it. The price of the software is essentially a fixed licensing fee per a copy paid to the owner. That is why, if the programmers are organized in a firm, they are involved in a group project and share the profits from the sales of the software they have created according to a certain apportionment rule. It is possible to assert that monetary incentives apply in this case, but agency costs are also present due to high monitoring costs and inefficient assessment of the individual effort. Holmstrom and Milgrom (1994) have found that "when the cost of measuring the performance is high, it is more likely that the agent's optimal incentives will conform with the attributes of employment."

b) Open Source Software Development

In the case of the open source, the situation is quite different. Programmers contribute their modules (or parts of a module) to a common pool, and the most active members by voting or the authority of a leader (informal project manager) individually choose the best version to become an "official" release. Since the final product is freely available, it is impossible to charge a positive price for it. The property rights in this case are widely dispersed within the network, and programmers do not participate in profit sharing. As Bessen (2001) claims, that "this appears to counter the common economic intuition... that agents without property rights, invest sufficient effort in the development of public goods" Bessen's argument descends to Arrow's (1962) work who argued that with information goods, under-investment may occur, because each agent faces a free-rider problem or pub-

lic externality. Coase (1960) argued that property rights allow *firms* to internalize these externalities, if transaction costs are zero.

c) Public good argument

The question is, however, whether the open source software *is* a public good. A public good is defined as one that is non-excludable and non-rival. However, as I mentioned above, the software is not a simple commodity, but rather a product with many features valuable differently for different customers. Not all the customers are going to use all the functions available, not all of them are even capable of doing it (for instance, reading and correcting the source code is only available for programmers, whereas certain features in a specialized program may be fully appreciated only by professionals in the field). Bessen (2001) refers to software as to a "complex public good" and provides a model, which implicates that property rights fail to internalize the public good externality with complex products, whereas users involved into development exploit private information to create a product with the features they most like or need. This leads us to the first incentive instrument used in OS development: *a need for a particular feature*.

Johnson (2000) proposes another view at the open source, where he considers OS as a public good, and he shows, quite in line with the argument above, that open source developers will tend to underprovide the effort waiting until other developers create the needed product first. However, Johnson does not personalize programmers' incentives. In his model, developers decide whether to produce or not based on the individually known costs and valuations associated with the development of a particular piece. Developers choose individual strategies by maximizing their expected payoff with respect to their beliefs about other developers. In the model, costs and valuations are impersonally mapped into strategy set, without detailization. In his approach, the incentive to develop is also a need for a particular function; however, free riding problem exists.

Garzarelli (2002) proposes to view the software as a *club good* in Buchanan's (1965) definition. Buchanan defined club goods as those lying between public and private goods. This is a club good in a sense that it is possible to naturally restrict "membership" in the club (to exclude non-experienced users unable to participate in development process). The author refers to the study by Antonelli and Foray (1992) who claimed that deve-

¹ It is possible to argue, that some of OS products are sold on the market for a positive price, e.g. various Linux distributions (Red Hat, SuSe, Debian, or Mandrake). However, it is plausible to assume that the price is charged not for the products on the distribution CD's, but rather for services attached to it: installation, support, maintenance, customization, etc.

lopers would cooperate in club only if the benefits of cooperation outweigh the costs. Here, the benefits of cooperation may be regarded to as information exchange and technical advice. A standard result of club theory is the decrease in the provision of the good when the number of users decreases, because significant *congestion* costs will accrue (I also considered this issue in my earlier work Vakhitov (1997)).

d) Network argument

If the OS software is considered as public good (with different modifications), the basic incentive of the developers is the need of the product and information exchange. However, externalities and free riding still persist, whereas participants of the Free Software Movement do not admit that this problem exists. Many OS researchers believe that this development model possesses very strong network effects. The network effect assumes that utility and benefits of everybody involved increase with the number of participants (network members). In some respect, the network argument contradicts the club good argument about the congestion. It is possible to explain this phenomenon with better access to information, which is non-congestable good, but another issue becomes important: the peer evaluation process.

Lerner and Tirole (2000) claimed that an agent engages in a certain activity if net benefits exceed net costs. The treatment of the costs was standard: time, effort and opportunity costs, but the benefits were split into immediate benefits and delayed benefits. The intermediate benefits include monetary compensation (if a programmer works for a firm), a needed bug fix or a piece of software, and personal enjoyment. The delayed benefits are the career concern incentive and ego gratification incentive. The former may lead to more desired job offers in the future, whereas the latter one stems from a desire for peer recognition. The authors refer to both incentives simultaneously as to a *signaling incentive*. Holmstrom (1999) considered conditions when the signaling incentives are stronger:

1. Visibility of the performance to the relevant audience. Since the source code is readily available for the peer preview, and looking through the code is the only way to determine its quality, a high quality code may earn a significant reputation among the peers. Besides, an early good project at the initial stage may attract new programmers and later on set a standard in the industry, that is relevant to ego gratification.

2. High impact of effort on performance. Since a developer acts independently, he is not bonded with relationships with a supervisor. Instead, he may allocate his time more efficiently. This argu-

ment is similar to delegating a part of firm's assets to an employee's possession. In our case, a valuable time serves as these assets.

3. Less idiosyncratic human capital. The software development process uses similar tools to build different blocks of the code. This implies, that once a programmer acquired some knowledge on how to perform a specific task, he may apply this knowledge to a wide array of similar tasks. In this respect, working on one's own as a part of OS project gives much more freedom than working in a team requirements of a firm.

These factors are more significant in OS projects, and so the signaling incentives. Thus, a desire to build a strong reputation may be a better incentive in the OS environment. Kelty (2001) comes up with a similar idea about the reputation as a strong incentive and enumerative for labor efforts, however, he built analogies with the gift economies.

e) Innovation argument

Creating a new code may also be considered as innovation process. Harhoff *et al.* (2000) provide incentive in use-developer innovating systems.

- Inducing manufacturer improvements, when innovation spills over to other users with the facilitating of a manufacturer (in our case it a person, capable of writing a code).
- Setting a standard advantageous to the user innovator. If a user comes up with a certain successful solution, it is in his strategic interests to reveal this innovation so that other users would also adopt it and it becomes a standard de-facto. In the IT industry, many to-day standards emerged from open source solutions.
- Reciprocity and reputation effects. When a user reveals a helpful innovation, other users may feel obliged to help him either. This might be an explanation of a high return rate in open source projects and their sustainability, because some developers contribute because they have already used open source solutions in their past.
- Low rivalry conditions. When the markets are disconnected, there is no a feeling that other users take an advantage from one's innovation. In the OS case this explains, why participants are not very concerned about free riding.

To summarize, the major incentives behind open source software development are the building of reputation, independent regime of work, and different ego gratification factors, such as creating a useful product for the community or simple personal satisfaction when the work is completed. These are the major theoretical findings in the recent literature. Does the empirical literature confirm or disprove those findings?

Empirical research in the OS development

The most papers dealing with OS are entirely theoretical and at most include a simple simulation results. To the best of my knowledge, empirical works in the field still have to be written. However, there are a number of studies, which summarize basic facts about the open source movement. These studies are based on various surveys of the developers' community. For an illustration, I have taken one of such surveys, conducted by Boston Consulting Group and presented in January 2002. The "Hacker Survey" was sent to randomly chosen 1570 developers involved in OS, and a bit more than 500 responses came back. Among many questions on the survey, the one we are interested the most is "What are your motivations in participating in OS movement?" The whole set of hackers could be roughly divided in four major categories:

1. *Believers* (33 %), who do it because they believe source code should be free.
2. *Professionals* (21 %), who do it for work need and professional status.
3. *Fun Seekers* (25 %), who do it for non-work need and intellectual stimulation, and.
4. *Skill Enhancers* (21 %), who do it for skill improvement.

As one can see, our analysis captures almost all of these categories, except the first one, that is probably may not be explained by economic rationale. Reputation building and ego gratification (personal pleasure, skill enhancement) are among the strongest incentives indicated. This confirms the fact that signaling incentives are strong in the community.

Professionals also show that they do it because their work [informally] requires so (70.8 %), to gain a professional status (45 %) or a status within OS community (22.6 %). Other categories reported much lower percentages for work functionality or work requirements. On average, 25 % to 38 % of respondents have shown that they feel obligation from use, that also confirms theory of innovative user-developers.

To conclude, the development of the open source is still a new and not very developed field. A significant effort should be pursued to discover some of the most vital questions. Among these questions are:

- What are motivations of OS developers, and whether they differ systematically among individuals?
- Do these motivations change over time and how?
- Under what conditions do individuals choose to join or leave the OS community?
- What are the individual costs and benefits of developers, and how it is possible to evaluate them?
- What are the governance structures within the community?
- Why do firms employ workers to participate in OS activities?
- Why don't all firms switch to OS software products? Are there switching costs?
- What is the support mechanism in the OS communities for inexperienced users? Is it sustainable in non-commercial setting?

These and many other questions are on the research agenda of those few who decided to devote their time and effort to studying the mechanisms and implications of the open source.

Alchian A. and Demsetz H. Production, Information Costs, and Economic Organization // *American Economic Review*- 1972.-V. 62(5).- P. 777-95.

Antonelli C. and Forray D. The Economics of Technological Clubs, Economics of Innovation and New Technology- 1992- V. 2.- P. 37-47.

Arrow K. Economic Welfare and the Allocation of Resources for Invention in *Nelson, R.* (ed.). *The Rate and Direction of Inventive Activity: Economic and Social Factors*, Princeton.- N. Y.: Princeton University Press. For the National Bureau of Economic Research, 1962,-P. 609-25.

Baker G., Jensen M. and Murphy K. Compensation and Incentives: Practice vs. Theory // *The Journal of Finance*- 1988.- V. 43(3). P. 593-616.

Behlendorf B. Open Source as a Business Strategy / In *DiBona C., Ockman S. and Stone M.* (eds) *Open Sources: Voices from the Open Source Revolution*.- Sebastopol: O'Reilly, 1999.

Bessen J. Open Source Software: Free Provision of Complex Public Good / Unpublished working paper.- 2001- Available at: <http://opensource.mit.edu>

Buchanan J. An Economic Theory of Clubs // *Economica*, NS- 1965.-V. 32(125). P. 1-14.

Coase R. The Nature of the Firm // *Economica*- 1937- V. 4(4).- P. 386-405.

9. *Coase R.* The Problem of Social Cost // *Journal of Law and Economics*- 1960-V. 8-P. 1-4.

10. *Garzaretti G.* Open Source software and the Economics of Organization, forthcoming in *Birner J. and Garroustle P.* (eds). Title Unknown-2002-Also available at: <http://opensource.mit.edu>

11. *Harhoff D., Henkel J. and Eric von Hippel.* Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations MIT Sloan School of Management WP.-2001.-№4125.

12. *Holmstrom B.* Moral Hazard in Teams // *Bell Journal of Economics*.- 1982.-V. 13(2).-P. 324-340.

13. *Holmstrom B.* Managerial Incentive Problems: A Dynamic Perspective // *Review of Economic Studies*.- 1999.- V. 66.- P. 169-182.

14. *Jensen M.* *Foundations of Organizational Strategy*.- Cambridge, MA: Harvard University Press, 1998.

15. *Johnson J.* Some Economics of the Open Source Software, Unpublished working paper.- 2000.- Available at: <http://opensource.mit.edu>

16. *Kelly C.* Free Software / Free Science.- 2001- First Monday, 6(12), available at http://www.firstmonday.org/issues/issue6_12/keity/index.html

17. *Klein B., Crawford R. and Alchian A.* Vertical Integration, Appropriate Rents, and the Competitive Contracting Process // *Journal of Law and Economics*,- 1978.-V. 21(2).-P. 297-326.

18. *Lerner J. and Tirole J.* The Simple Economics of Open Source. NBER Working Paper-2000.-№ 7600.
19. *Milgrom P. and Roberts J.* Economics Organization and Management. Englewood Cliffs-N. Y.: Prentice Hall, 1992.
20. *Raymond E.* The Cathedral and the Bazaar- Sebastopol: O'Reilly Press, 2001.
21. *Simon H.* The Formal Theory of the Employment Relationship //Econometrica- 1951- 19(3).- P. 293-305.
22. *Vakhitov V.* Internet Pricing Models under Congestion,- Unpublished MA Thesis, 1997.
23. *Williamson O.* The economic institutions of capitalism: Firms, markets, relational contracting.—N. Y.: Free press, 1985.

В. Вахімов

СИСТЕМИ МОТИВАЦІЇ ЩОДО РОЗВИТКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВІДКРИТИМ КОДОМ

У статті розглянуто різні економічні теорії розробки програмного забезпечення з відкритим кодом, покликані відповісти на досить просте запитання: навіщо численні користувачі й розробники витрачають велику частину свого часу та значні зусилля, створюючи фактично безкоштовний продукт загального користування? Окрім стандартного ключа, запозиченого з економічної теорії фірми, розглянуто також нові теоретичні підходи, накреслені автором спеціально для того, щоб пояснити феномен створення саме таких програм.