1K08 How innovative is open source software? Empirical analysis

○KLINCWEICZ, Krzysztof, MIYAZAKI, Kumiko (Tokyo Institute of Technology)

The paper addresses an ongoing debate about the innovativeness of open source projects and critically evaluates the innovative potential of 500 most active projects registered by SourceForge.net. The analysis is based on a proposed framework, distinguishing between radical inventions, technology / platform modifications, and marketing innovations. Research findings include relatively low levels of technical newness in the studies sample, contrasting with high interest of developers and users in the innovative projects. The article discusses the underlying mechanisms, restricting innovativeness of community-driven open source efforts, and postulates the establishment of an institution of "idea brokers", playing roles corresponding to venture capitalists in the commercial software domain.

Open Source Software and innovation

Imagine an inventor, who has a promising idea, which could be turned into a potentially successful software application, but this would require cooperation with experienced programmers. Can the open source software (OSS) community offer him compelling mechanisms, helping implement the innovation? If so, what are the prospects for the OSS to displace the current commercial efforts of established software companies – is it an alternative value creation mode, or is its current role restricted to commoditization of established technological designs?

The popular view of open source movement postulates that it leads to faster incorporation of innovative ideas than the proprietary regime, pursued by commercial companies (Tuomi 2005: 434). The innovative opportunities are however not available to every open source project, and development efforts for many promising applications have been suspended because of insufficient programmer contributions. Selection and mortality rates of OSS projects are higher than those of commercial initiatives, which are often maintained because of existing user bases, reputation effects, or sunk costs. The above view of OSS innovation seems therefore idealistic – many projects end up generating yet another "me-too" product, focused on usability, support for a specific platform and commoditization of features, so far available from commercial vendors, and there are only few open source projects venturing into new, previously unexplored areas.

Many researchers tend to classify every open source product as an innovation, thus using the terms "open source development" and "open source innovation" synonymously (comp. Harhoff, Henkel, et al. 2003; Hippel, Krogh 2003; Krogh, Spaeth, et al. 2003). This confusion results in a distorted image of the OSS community – the implicit assumption of the innovativeness of every OSS project leads to the use of value-bound statements in definitions of research problems, and the term "innovation" faces the risk of turning into a buzzword in the context of the software industry. The biased interpretation of OSS innovativeness contrasts with other approaches, attempting to introduce a distinction between imitation and actual novelty – as Tuomi noticed, apart from the unique development model, "there is nothing particularly innovative in projects such as Linux, which basically reimplements commercially available operating-system functionality" (Tuomi 2005: 436), and some authors prefer to enumerate certain innovative achievements of the OSS community, intentionally not generalizing the term to cover all possible developments.

Innovations in the software industry

The question of innovativeness calls for a better understanding of the term innovation, especially in the context of the software industry. Roger's seminal work on the diffusion of innovations defined an innovation based on its perceived newness to users (Rogers 2003: 12), and his followers emphasized that "as long as the idea is perceived as new to the people involved, it is an «innovation», even though it may appear to others to be an «imitation» of something that exists elsewhere" (Van de Ven 1986: 592) - paradoxically, products presented as innovative no longer need to be new and unique.

«New» turns out to be a broad term in marketing literature, referring to three categories of products: new to the world (with no comparable alternatives available), new to the firm, or improved (through changes to an existing product) (Kamel, Rochford, et al. 2003). Apart from this "market newness", one can also identify technical product newness, when products are based on newly developed technologies, their unique combinations, or original applications of a well-known technology (Loch 2000: 257). Creative re-positioning of technically unchanged products (*marketing innovation*) is also an effective mechanism for improving sales in the software industry (Klincewicz, Miyazaki 2004) – but aesthetic, usability or conceptual changes are strikingly different from the actual technology development. Innovative products encompass therefore categories as diverse

as: radical innovations (with new technologies addressing new markets or user needs), line extensions (where established technical designs are used to serve new customer segments), and incremental projects (Loch 2000: 249) – the term "product innovation" equals "product diversification". The fascination with innovations poses also a more general problem, identified as "pro-innovation bias", an implicit assumption that an innovation should be diffused and adopted by all potential users (Rogers 2003: 106), causing particular confusion in the software and IT services markets.

The present study tries to return to the intellectual roots of the concept of innovation by distinguishing between original and "me-too" products. Some authors made attempts to identify criteria differentiating radical innovations from less important changes – a notable effort by Dahlin and Behrens (2005) offered operationalized criteria and verification techniques. Radicalness results from technical contents of a product, not its diffusion, and a radical invention was defined as (Dahlin, Behrens 2005: 725):

(1) novel (dissimilar from previously available inventions);

(2) unique (diverging from current interests of other inventors);

(3) having an impact on future technologies (encouraging imitation).

The novelty and uniqueness criteria are satisfied only if no functionally comparable products exist at the launch date of a development project.

Application of the framework to software products calls for an additional modification: software applications are developed for specific underlying platforms, and in most cases, an application built for one platform cannot easily be ported to another system. The interdependencies resulting from technological standards stimulate the formation of platforms (Cusumano, Gawer 2002), where individual software firms do not compete directly against supporters of incompatible platforms. In the OSS community, many development efforts are focused on improving and complementing the Linux platform. Even though certain applications exist for Windows users, they need to be "re-invented" for the other operating system environment. These re-inventions are not radical technological breakthroughs, as similar benefits and functionality are already available for alternative platforms. If their functionality is new for a specific platform, they could nevertheless be regarded as "local scale radical inventions" and will later be referred to as "platform modifications".

Consequently, in order to capture the complexity of innovations in the software industry, the following types of software innovation are distinguished:

- radical inventions (breakthroughs) – products new to the world,
- technology modifications – significant incremental improvements of established technologies,
- platform modifications – products so far available only for competing platforms,
- marketing innovations – resulting from original positioning and new uses of existing technology solutions.

Apart from product-related innovations, OSS movement generates also numerous process and organizational innovations, which are not included in the present study.

Research methods

The present research analyzes the innovativeness of OSS projects, registered in SourceForge.net - the most comprehensive portal for open source developers. In July 2005, the portal had over 100,000 projects and over 1 million registered users. Due to its comprehensive coverage and querying potential, SourceForge was a popular source of data for research concerning the OSS community (Weiss 2005; Hahsler, Koch 2005; Crowston, Howison 2005). The present study analyzes a large-scale sample of 500 OSS projects, selected from SourceForge list of projects with the highest activity (activity levels are automatically registered by the portal based on additions of messages, modified code and other development efforts), and was facilitated by *tech mining* software VantagePoint. Even though SourceForge is the most representative collection of OSS projects, it does not cover all relevant open source communities. Certain visible projects maintain independent developer websites, and many highly active projects use SourceForge merely as a source code repository, redirecting all other traffic (including support requests) to own websites – this results in relatively low activity statistics registered.

	Population	**Selected sample**
Number of projects	17,139 (*only active)	500 (3%)
Number of developers	34,393	6,106 (18%)

Table 1: SourceForge.net statistics (data for the entire population from: Weiss 2005)

Research findings

Most of the analyzed projects rely on relatively small teams of developers: 9% of them are maintained by one person only, and 22% by 2-4 developers. The average number of developers per project is however 12.21, significantly more than the mean of 2.0067, computed for the entire SourceForge population (Weiss 2005: 31). Surprisingly, all of the analyzed projects were registered between the years 1999 and 2001 – none of the projects started in the following four years gained enough popularity to join the most active group.

Table 2 presents the results of the analysis, which divided the 500 projects into distinctive groups, based on the degree of newness. Only 64 projects (12.8%) were not direct imitations of other existing solutions. Descriptions of the 436 remaining projects, classified as non-innovative, frequently use words such as "similar", "one of", "enhancement of/based on [another software]", "yet another", "replacement". Redundancy is a serious problem: "yet another"

application offering similar functionality creates unnecessary competition for scarce resources within the developer community, reducing their effective utilization. In the OSS world, mergers and acquisitions do not occur, and joining forces by members of competing projects is a rare phenomenon – whereas among commercial software companies, similar product lines from several vendors may eventually merge into one system as a result of alliances, take-overs or technology licensing agreements.

	New technology	New for a platform	Existing technology
New market	**Radical invention (breakthrough) 5 (1.0%)**		**Marketing innovation 3 (0.6%)**
Existing market	**Technology modification 4 (0.8%)**	**Platform modification 52 (10.4%)**	**No innovation 436 (87.2%)**

Table 2: Types of product innovation among the 500 most active SourceForge projects

The findings reveal low overall share of unique OSS projects - as Table 2 indicates, only 5 out of 500 SourceForge projects could be classified as technological breakthroughs, 4 were technology modifications, 3 - marketing innovations, and the remaining 52 projects were platform modifications, focused on implementing functionality new to a specific platform, but already available for other, incompatible systems. Among platform modifications, Linux is the leading environment (contrary to the high Windows' popularity, registered for the entire sample), and projects in this group port new functionality, popular among Windows users, or improve support for specific standards and hardware (e.g. USB, PMCIA, NTFS, IEEE 1394). Such projects are characterized by a "self-service" mode of development: end users "tweak the hardware support" for their own systems in a way that resembles the concept of *prosumption*, where consumers assume responsibility for the production of consumed goods (Toffler 1990). In spite of the disappointing results of the survey, truly innovative OSS projects exist, but many of them were implemented by individual programmers, and not listed by SourceForge. In the OSS community, there seem to be limited prospects for the division of labor between inventors and implementors: if you have a new idea, or need specific complementary (and yet unavailable) functionality, you have to implement it by yourself. Pioneering new concepts is more difficult than implementing and commoditizing proven designs, developed for other platforms by commercial companies.

Table 3 presents comparative statistics for various types of projects, in two cases excluding SourceForge project (development of the portal is also an OSS project, critical for all portal users and therefore different from other open source efforts). New feature requests represent suggestions and ideas coming usually from software users, support requests concern problems with using a product, while other messages refer to various organizational and technical aspects of development projects. Innovative projects are significantly more popular among developers than "me-too" solutions. Development of underlying technology platforms attracts more attention than platform modifications.

Stimulation of innovation is an additional challenge for the OSS community. Follow-up content analysis of the documentation of the identified innovative projects revealed (Table 4) that 40% of breakthroughs came from company-initiated projects, and 50% of technology modifications grew out of academic research, while community-driven initiatives were in turn more focused on platform modifications and marketing innovations.

Project type	Average no. of developers	Average no. of new feature requests	Average no. of support requests	Average no. of messa-ges
All	12.21	79.47	207.84	902.13
All (excluding SourceForge)	12.19	74.08	32.56	895.56
Non-innovative projects	12.30	77.98	33.42	921.86
Radical inventions	21.40	555.80	17542.20	1317.00
Radical inventions (excluding SourceForge)	85.00	10.00	39.00	2403.00
Technology modifications	38.50	236.25	0.00	0.00
Platform modifications	8.79	35.83	31.65	795.23
Marketing innovations	7.67	28.67	1.67	228.00

Table 3: Statistics for different types of projects

Project type	Companies	Academia	Community
Radical inventions	2 (40.0%)	0	3 (60%)
Technology modifications	0	2 (50.0%)	2 (50.0%)
Platform modifications	5 (9.5%)	1 (2.0%)	46 (88.5%)
Marketing innovations	0	0	3 (100%)
All innovative	7 (10.9%)	3 (4.7%)	54 (84.4%)

Table 4: Initiators of innovative projects

New functionality is often offered as "feature gifts", original code developed by individuals or companies outside the OSS community and contributed upon joining a project (Krogh, Spaeth, et al. 2003: 1233) – but such inventions are not really generated within the OSS development model itself. Similarly, "source code opening" means that a commercial company decides to share its innovations and let the community

maintain and modify them. Many companies use OSS licensing model and developer communities, while pursuing parallel commercialization strategies, and their innovativeness is linked to clear financial motives, not present in typical OSS projects. One should therefore not confuse efforts, where open source licensing is incidentally used, with truly community-driven developments: they differ in technology control modes, decision making processes about development directions, and motivations to innovate.

Open source projects often originate from university research or hobbies of individual programmers – many of them implement novel ideas, but at the same time have limited practical uses outside narrowly defined fields of interest. In the case of university research, the application of the open source licensing does not need to be combined with a real community-driven development model, as the actual research is financed from other sources, and driven by objectives not related to the OSS community. Many ideas coming from academics are interesting, and could be inspiring for other developers – but they usually do not reach the developer community, which lacks open source counterparts of "technology transfer" mechanisms, regulating relations between academia and commercial companies.

Discussion

Nobody keeps track of innovative open source projects, abandoned by their founders, who were not able to find sufficient support from other developers. Lack of diffusion prospects for innovations may discourage also their generation, and pioneers are inclined to look for alternative ways to implement their ideas. The analyzed statistics of OSS projects indicated relatively low levels of innovativeness. Large, established technology companies can afford the luxury of limited innovativeness, acquiring new ventures to extend own product portfolios – in contrast, community movement can only generate new value by own development, or by convincing commercial vendors to open code of of their existing applications and donate "feature gifts".

The above discussion should not lead to a conclusion, that the open source movement is entirely stripped of innovative capabilities. There have been truly innovative projects initiated by the OSS community, but the actual problem is not the inflow of ideas, inputs to new product development: it is the ability to promote new ideas, gain support from other community members and stimulate the diffusion of new applications. While OSS developers have excellent tools for software engineering, technical support groups and mechanisms, stimulating code re-use to shorten development cycles, they lack efficient project promotion frameworks. Commercial software developers can resort to venture capitalists to receive funding, as well as marketing and networking support. The OSS community desperately needs corresponding "idea brokers", helping launch innovative initiatives, which do not necessarily fit into existing mainstream developments. Due to the nature of open source movement, funding is not as important as an independent evaluation of concepts, business consultancy (including advices on user needs, project positioning and potential partnerships with other projects), and promotion of promising projects to create distributed developer communities. This role could be taken by a dedicated open source foundation, actively seeking sponsors for these activities. Alternatively, one of large IT companies, showing its commitment to the OSS community by opening often low-end, obsolete technologies and expecting to have their code maintained and improved, could instead demonstrate support for truly innovative and open initiatives.

Literature

Crowston, K., Howison, J. (2005) The social structure of free and open source development. *First Monday*, 10, 2, www.firstmonday.org/issues/issue10_2/crowston

Cusumano, M.A., Gawer, A. (2002) The Elements of Platform Leadership. *MIT Sloan Review*, 43, 3, pp. 51-58

Dahlin, K.B., Behrens, D.M. (2005) When is an invention really radical? Defining and measuring technological radicalness. *Research Policy*, 34, pp. 717-737

Hahsler, M., Koch, S. (2005) Discussion of a Large-Scale Open Source Data Collection Methodology. Proceedings of the 38th *Hawaii International Conference on System Sciences*

Harhoff, D., Henkel, J., Hippel, E. von (2003) Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research Policy*, 32, pp. 1753–1769

Hippel, E. von, Krogh, G. von (2003) "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*, 14, 2, pp. 209-223

Kamel, M., Rochford, L., Wotruba, T.R. (2003) How New Product Introductions Affect Sales Management Strategy: The Impact of Type of "Newness" of the New Product. *The Journal of Product Innovation Management*, 20, pp. 270-283

Klincewicz, K., Miyazaki, K. (2004) Dilemma in Innovation. The Case of Product Innovations versus Marketing Innovations in the Software Industry. *The Japan Society for Science Policy and Research Management Yearbook*, Tokyo, pp. 107-110

Krogh, G. von, Spaeth, S., Lakhani, K.R. (2003) Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32, pp. 1217-1241

Loch, Ch. (2000) Tailoring Product Development to Strategy: Case of a European Technology Manufacturer. *European Management Journal*, 18, 3, pp. 246-258

Rogers, E.M. (2003) *Diffusion of innovations*. Free Press, New York

Toffler, A. (1990) *The Third Wave*. Bantam Book, New York

Tuomi, I. (2005) The Future of Open Source, [in:] Wynants, M., Cornelis, J (2005) *How Open is the Future*? VUB Brussels University Press, Brussels, pp. 429-459

Van de Ven, A.H. (1986) Central Problems in the Management of Innovation. *Management Science*, 32, 5, pp. 590-607

Weiss, D. (2005) *A Large Crawl and Quantitative Analysis of Open Source Projects Hosted on SourceForge*. Poznan University of Technology, Poland, RA-001/05, www.cs.put.poznan.pl/dweiss/site/publications/download/weiss-2005-large-crawl-of-sourceforge.pdf