

# Assessing Innovation in the Software Sector: Proprietary vs. FOSS Production Mode. Preliminary Evidence from the Italian Case

Dario Lorenzi and Cristina Rossi  
Department of Management, Economics, and Industrial Engineering  
Politecnico di Milano

**Abstract.** Innovation in the software sector is a widely debated issue. Which are the most important dimensions to assess innovation in this field? Can we measure innovative processes carried out by software companies and what kind of innovation do they develop? Are FOSS solutions more innovative than proprietary ones? These are the research questions we endeavor to answer in this paper providing some empirical evidence, obtained using an original methodology of collecting experts' evaluations on the innovation level of 134 solutions provided by a group of Italian Small and Medium Enterprises.

**Keywords:** Innovation, Free/Open Source software, proprietary software

## 1 Introduction

The issue of innovation in the software industry is of great interest for both academic and practitioners (Bloch, 2007). Which is an innovative software solution? Which aspects should be considered to highlight the most important elements of innovation processes?

In this framework, the success of Free/Open Source Software (FOSS) highlights new research issues, dealing with whether and how FOSS programs turn out to be more innovative than proprietary ones (Klincewicz, 2005). The theme is intriguing as FOSS represents a disruptive process innovation affecting industrial dynamics (Dalle et al., 2007), and it can be regarded as an important instantiation of the Open Innovation model (West and Gallagher, 2006).

We focus on Italian software sector, addressing three research questions: (i) are software solutions developed by Italian SMEs' innovative? (ii) What kinds of innovations are implemented? (iii) Are FOSS programs more innovative than proprietary ones?

---

*Please use the following format when citing this chapter:*

Lorenzi, D. and Rossi, C., 2008, in IFIP International Federation for Information Processing, Volume 275; *Open Source Development, Communities and Quality*; Barbara Russo, Ernesto Damiani, Scott Hissam, Björn Lundell, Giancarlo Succì; (Boston: Springer), pp. 325–331.

## 2 Review of Literature

The concept of innovation is one of the mostly studied by economic scholars (Fagerberg, 2004). Particularly, it was acknowledged the complexity of the phenomenon and how to measuring it turns out to be a critical task. A short list of the most used metrics of firms' innovation effort include: (i) data on Research and Development; (ii) patents; (iii) bibliometrical data, such as publications in scientific and technical journals.

All previous indicators suffer from several shortcomings (Kleinknecht et al., 2002), in particular as far as the sectors forming the so-called *new economy* (Haskel, 2007). Indeed, the passing from a commodity-driven to a knowledge-driven economy forces to re-define innovation, considering a whole new set of variables related to how knowledge is created, managed, and passed through different actors (Bloch, 2007). Specifically, it has been noted that patents are often unable to follow the rapid evolution of the software market and to account for the complexity of complementary activities (Blind et al., 2004).

As a consequence, new indicators are needed to assess innovation in ICT fields (Maruyama et al., 2007). In the case of software, such indicators should take into account not only general aspects, but also specific elements (as the use of certain programming languages and platforms and so on).

In this framework, the rapid pace of diffusion of FOSS represents another source of complexity. Klineciewicz (2005) has attempted to provide a classification of innovation in FOSS, basing on four classes: radical innovation, technological modifications, platform modifications, and new uses for existing technologies. The authors have also analysed 500 projects hosted on SourceForge<sup>1</sup>, assessing that 436 of them are not innovative. This result can be linked to other studies showing low confidence in the innovative impact of FOSS solutions (Tuomi, 2005). In brief, the debate is far from a conclusion, even if FOSS solutions have proved to be as complex and reliable as proprietary ones.

## 3 Data and Methodology

The sample comes from a survey taken, in 2004, on more than 900 software firms<sup>2</sup> from Finland, Germany, Italy, Portugal, and Spain (Second European Libre Software Survey, ELISS II, see Bonaccorsi et al., 2006). Specifically, we focused on the 323 Italian respondents, because of database characteristics (some information are not available for all countries) and peculiarity of this software market, shaped by the presence of Small and Medium Enterprises (SMEs). More than 90% of firms have a total staff of less than 20 employees, and, in 16% of cases, they are one-man businesses. 167 of the companies provide to their customers FOSS-based software.

---

<sup>1</sup> <http://www.sourceforge.net>

<sup>2</sup> NACE code 72.2.

The average percentage of graduate staff is fairly high (about 36%), so as that of software developers (almost 30%). Respondents serve mainly business customers (81%), particularly SMEs (64%), while very few refer to University (3%) or end users (3%).

In order to address whether traditional innovation indicators are really not suitable for assessing innovation, data on three main metrics have been collected: (i) trademarks, using the database of the European Office for Trademarks and Design; (ii) patents, referring to the Delphion database (); (iii) scientific and technical publications hosted on Scopus<sup>3</sup>. Methodology is based on evaluations by software experts, basing on information recorded through the Web sites of the 323 Italian respondents. We collected comprehensive information on 134 solutions developed by a subsample of 70 companies. It is important to underline that, according to this methodology, the unit of analysis is no more the firm, but its software solutions. The 134 programs target mainly business customers: only 8% are intended for home users, and the most widespread category is managerial systems (45%), followed by Web oriented applications (9%). Moreover, most of them (107) are released under proprietary license, while the remaining ones (27) are distributed under FOSS licenses.

Three experts<sup>4</sup> were asked to evaluate the level of innovativeness: after an in depth colloquium, they received a table (containing the name of each product, the link to its Web page, a brief description of the product and of the producing firm) to be filled and a guide for its compilation. It is worth noting that, in order not to introduce a bias, we did not tell the experts whether a software was FOSS or not; anyway, they might find this information on the software Web site. Evaluators were asked to assign a mark between 1 (*not innovative*) to 5 (*very innovative*) to each product, referring to three main dimensions, leading to a set of five indicators. The choice of the dimensions is driven by economic literature on the topic (see, for instance, Garcia and Calantone, 2002).

The first dimension is related to the internal level of innovation, comparing each software with other programs developed by the same company (*Indicator 1: is the product new to the firm?*). The second one refers to the market in which the firm operate, comparing each product with similar available solutions. This dimension is addressed by two indicators, related to innovation in *what* a software does (*Indicator 2: is the software innovative in the sense that it better satisfies needs from users?*), and in *how* it succeeds in accomplishing a task, exploring technical aspects (*Indicator 3: is the product technologically new to the market?*).

The third dimension addresses innovation referring to the general level of technology and knowledge (*is the product new to the world?*). Two indicators are developed: (i) innovation in modules composing the software (*Indicator 4: is the software innovative as it contains original modules, which can be hardly found in other solutions, also in different market segments?*); (ii) innovation in what we labelled as *platform*, meaning those aspects related to the other technological characteristics, such as

<sup>3</sup> <http://oami.eu.int>, <http://www.delphion.com>, <http://www.scopus.com/scopus/home.url>

<sup>4</sup>Our decision to choose these experts was driven by geographic closeness and budget considerations. We are confident that they are well qualified to evaluate these products, having a long working experience in the software field.

programming languages, implemented algorithms, use of libraries, and so on (*Indicator 5: how the software platform is new with respect to those of other software solutions, also in different market segments?*).

It is useful to relate our indicators to dichotomy between *radical vs. incremental innovations*. For instance, innovations more related to technical aspects (indicator 3 and 5) can be considered as more radical innovations, with respect to a solution having original modules as the only element of novelty.

The main shortcoming of this methodology is the subjective nature of evaluations. However, all software have been evaluated by three experts: their knowledge of the market and multiple opinions mitigate the problem. Moreover, metrics of accordance of their judgement have been computed, showing that they were correlated<sup>5</sup>. Concluding, we chose to compute, for each indicator, the sum of the three evaluations, obtaining scores ranging between three (all experts assigned one) and fifteen (all five), and preserving variability even if mitigating the effect of outliers.

## 4 Empirical Results

Traditional metrics seem not to be suitable for the software sector formed. According to our investigations, only 5 companies registered, globally, 35 trademarks (26 logos and 9 names of companies). Only 3 firms hold a patent (for a total of 15 patents), and 3 were involved in scientific publications. In presenting experts' evaluations, we follow the repartition previously mentioned: innovation within the firm (dimension 1, indicator 1); innovation within the referring market (dimension 2, indicators 2 and 3); overall innovation (dimension 3, indicators 4 and 5).

Table 4.1: Descriptive statistics.

Dimension	Indicator	Min.	Max.	Std.Dev.	Median	75 <sup>th</sup>	90 <sup>th</sup> perc	99 <sup>th</sup>
I	1	4	14	2.11	8	10	11	14
	2	3	14	2.46	9	10.7	12.7	13.7
II	3	3	14	2.78	8	10	12	13.7
	4	5	13	2.04	8	10	11	13
III	5	3	14	2.46	8	9	11	13.7

For the first dimension, 99<sup>th</sup> percentile is 14 (the highest for all dimensions). Standard deviation is low, showing concentration around mean values (more than 50% of the sample is between 6 and 8). Two indicators (innovation in *what* a software does: *indicator 2*; innovation in technical aspects: *indicator 3*) capture the second dimension. Both show equal values for minimum and mode, while differences in percentiles emerge: in 75<sup>th</sup> and 90<sup>th</sup> percentiles, indicator 2 has higher values (respec-

<sup>5</sup> As our study is characterised by multiple ratings per subject, we used the Spearman's rank correlation coefficient.

tively, 10.7 vs. 10, and 12.7 vs. 12). This finding is confirmed by other analysis: 69% of solutions have received a mark of 7 or more for indicator 2, while the proportion decreases to 52% for the third indicator. The innovative process seems to be more effective for aspects related to what a software does, instead of technological aspects. Third dimension was distinguished between innovation in modules (*indicator 4*) and in other technical aspects (*indicator 5*). 75<sup>th</sup> is higher for indicator 4 (10 vs. 9), but this pattern is not still true when focusing on values closer to the upper limit (99<sup>th</sup> percentile: 13 vs. 13.7). Experts claimed that sometimes Web sites provided few information about modules: this have generated a *central tendency bias*, leading evaluators to assign score 3 (confirmed by standard deviation: 2.04 for modules vs. 2.46). Indicator 4 obtains higher scores: proportion of evaluations between 5 and 7 is predominant for indicator 5 (32% vs. 38%), while indicator on modules presents more values between 7 and 9 (53% vs. 43%). This may be related to the peculiar structure of the sample: in a market dominated by SMEs, it is likely to observe more customizations and adaptations than radical innovations based on new technologies.

Summing up, data allow us to answer the first two research questions. As far as the former (*are Italian software solutions innovative?*), whilst results with traditional instruments are useless, using a methodology based on experts' evaluations, we succeed in disentangling innovation into its main dimensions. Also the latter issue (*what typologies of innovation are implemented?*) receives an answer. On one hand, when focusing on the referring market, indicator 2 shows higher values than indicator 3, supporting the idea of innovation processes more targeted to *what* a software solution does, than to *how* to do it. On the other, considering the *new to the world* dimension, innovation seems to be more focused on modules than in technical aspects. According to our findings, the Italian software sector seems to be characterized mainly by adaptations, customizations, and transfer of solutions into different platforms (an innovation that can be appointed as incremental).

In order to address our third research question (*are FOSS solutions more innovative than proprietary ones?*), we have performed comparisons between these two groups. As mentioned in the previous section, the sample of 134 software was formed by 109 proprietary and 27 FOSS solutions.

Table 4.2: Descriptive statistics. Proprietary vs. FOSS solutions

		Proprietary solutions							FOSS solutions						Mann Whitney P value	Nonparametric equality of medians P value
Di m	Ind	Mi n	Ma x	Me-dian	Std Dev	75 <sup>th</sup> per	90 <sup>th</sup> per	Mi n	Max	Me-dian	Std Dev	75 <sup>th</sup> per	90 <sup>th</sup> per			
I	1	4	14	8	2.03	10	11	6	14	10	2.15	11	12	0.007	0.027	
	2	3	14	8	2.30	10	11	5	14	11	2.66	12.5	13	0.004	0.002	
II	3	3	14	7	2.55	9	11	3	14	11	3.14	12	13	0.003	0.001	
	4	5	13	8	2.03	10	11	5	13	9	2.12	10	11	0.428	0.061	
III	5	3	14	7	2.21	9	10	4	14	10	2.58	11	12.4	0.000	0.000	

FOSS solutions seem to be more innovative: in all the three dimensions, evaluations are higher for FOSS. Specifically, indicator 1 has higher median for FOSS and statistical tests show that this difference is significant. This is corroborated by data distribution: focusing on the score range 5-7, the proportion for FOSS (26%) is lower than the one for proprietary programs (44%), while the opposite emerges for values equal to 10 or more (51% vs. 27%). As far as the second dimension, FOSS solutions seem to be more innovative too. For both indicators, statistical tests confirm a higher level of innovativeness. Focusing on differences between indicators 2 and 3 within the same group, we notice that, in both cases, evaluations for indicator 3 are lower than those of indicator 2, confirming the conclusions for the entire sample. Same pattern emerges with the third dimension. However, whilst statistical tests confirm significant differences in median for indicator 5, no significant difference between the two sub-samples emerges for indicator 4, but, as mentioned above, results could be invalidated by central tendency bias. If the comparison between the two groups is interesting, it is also of interest to consider the differences between the two indicators within each group. Indeed, proprietary solutions show higher evaluations for modules, as emerged for the entire sample (e.g. values over 7 constitute the 65% for indicator 4, while only the 46% for indicator 5), while FOSS programs follow an opposite pattern. Indeed, indicator 5 shows higher values, as highlighted by median (9 for indicator 4 vs. 10 for indicator 5), 75<sup>th</sup> percentile (10 vs. 11), and 90<sup>th</sup> percentile (11 vs. 12.4). A more in depth analysis allows to notice that a large part for this difference comes from the highest values. This last consideration can be regarded as an insight that proprietary and FOSS software are also shaped by different innovative processes: radical innovation in FOSS vs. incremental innovation in proprietary field.

## 5 Conclusions

Innovation in software industry can be hardly defined and measured: this paper proposes an original methodology for assessing innovativeness of software solutions produced by Italian SMEs. We have provided evidence that innovative processes exist and it is possible to assess them under different perspectives and to delineate general characteristics of these processes, showing that innovations related to technical aspects are less prominent: it is possible to address these innovation processes as incremental, which is in agreement with a software sector shaped by SMEs. Moreover, our analysis has highlighted differences between innovativeness of FOSS and proprietary software. Specifically, FOSS solutions display higher level for all indicators, and almost all statistical tests indicate that the two groups are two different populations. Differences emerge in the level of innovation and in the relationships between indicators: specifically, focusing on the global level of technology, FOSS solutions show higher values for technical aspects. The characteristics of the sample made results hardly generalizable; however, they help in shed light on questions posed by the FOSS paradigm. Does the FOSS production foster innovation processes? Does the FOSS represent a valid alternative for software SMEs wishing to operate with lead-

ing-edge technologies? Hence, it would be interesting for future researches to apply a refined version of this methodology on larger database, containing data from different countries.

## References

- Blind K., Edler J., Friedewald M. (2004) *Software Patents – An Empirical Analysis from an Economic Perspective*. Stuttgart: Fraunhofer IRB Verlag
- Bloch C. (2007) Assessing recent developments in innovation measurement: the third edition of the Oslo manual. *Science and Public Policy*, 34 (1): 23 – 34
- Bonaccorsi A., Giannangeli S., Rossi C. (2006) Adaptive entry strategies under competing standards-hybrid business models in the Open Source software industry. *Management Science*, 52(7): 1085-1098
- Dalle J.M., Rossi C., Rullani F. (2007) Toward a new industrial organization? OSS in economic and managerial perspective. In Feller J., Fitzgerald B., Scacchi W., Sillitti A. (eds.) *Open Source Development, Adoption and Innovation*, Springer, New York, NY
- Fagerberg J. (2004) Innovation: a guide to the literature. In Fagerberg, Jan, David C. Mowery and Richard R. Nelson: *The Oxford Handbook of Innovations*. Oxford University Press: 1-26
- Garcia R., Calantone R. (2002) A critical look at technological innovation typology and innovativeness terminology: a literature review. *he journal of product innovation management*, 19(2): 110-132
- Haskel J. (2007) Measuring innovation and productivity in a knowledge-based service economy. *Economic and Labour Market Review*, 1 (7): 27-31
- Kleinknecht A., van Montfort K., Brouwer E. (2002) The Non-Trivial Choice between Innovation Indicators. *Economics of Innovation and New Technology*, 11 (2): 109-121
- Klincewicz K. (2005) Innovativeness of open source software projects, MIT online paper, Boston. Available at: <http://opensource.mit.edu/papers/klincewicz.pdf>
- Koberg S., Detienne D., Heppard K. (2003) An empirical test of environmental, organizational, and process factors affecting incremental and radical innovation. *The Journal of High Technology Management Research*, 14 (1): 21-45
- Maruyama, Kohda, Katsuyama (2007) Issues of Service Innovation and Its Model. The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services: 489-490
- West J., Gallagher S. (2006) Patterns of open innovation in Open Source software. In Chesbrough H., Vanhaverbeke W., West J., Eds. (2006) *Open Innovation: researching a new Paradigm*, Oxford University press, Cambridge