

# From “Community” to “Commercial” FLOSS – the Case of Moodle

Andrea Capiluppi, Andres Baravalle

Centre of Research on Open Source Software – CROSS  
University of East London  
London, UK  
[a.capiluppi, a.baravalle}@uel.ac.uk](mailto:{a.capiluppi, a.baravalle}@uel.ac.uk)

Nick W. Heap

The Open University  
Walton Hall, Milton Keynes, UK  
[n.w.heap@open.ac.uk](mailto:n.w.heap@open.ac.uk)

*Abstract*—This paper qualitatively documents how the decision of the largest UK e-learning provider to adopt Moodle had an effect on this FLOSS system. The study of its evolution provides evidence of the success of this project: increasing amounts of provided effort by developers correspond to similarly increasing produced outputs to the Moodle system. It becomes evident how commercial partners helped this OSS system to achieve the transition from an “average” OSS system to a successful multi-site, collaborative and community-based OSS project.

*Keywords*—evolution; FLOSS systems; phases

## I. INTRODUCTION

This paper studies the evolution of the Moodle e-learning platform, and describes how it achieved the status of “successful FLOSS project” thanks to the Open University (OU), the largest online course provider in the UK. In order to achieve this, this paper uses a mixed qualitative and quantitative approach, and uses a wealth of information sources, ranging from interviews with commercial stakeholders in Moodle, to empirical data contained in the Moodle code repository. It is argued that this system represents a “hybrid” OSS project [6]: since its inception in the early 1980’s, OSS projects were purely volunteer-based, heavily relying on personal efforts and non-monetary recognitions, and bearing communication and coordination issues (“Plain OSS”, right end of Figure 1, adapted from [6]). Nowadays Commercial OSS are also present (more similar to Closed source systems, as in Figure 1), where a commercial company plays a major role in the development and decision making. Community OSS instead are more similar to pure OSS systems, they are driven by the community, but they also often have several commercial stakeholders.

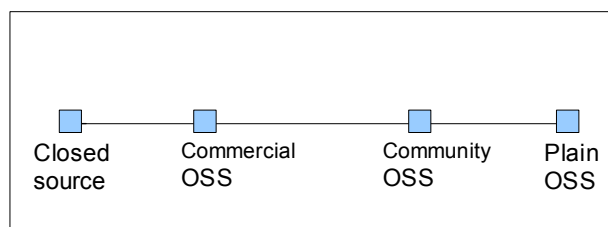


Figure 1: Software licensing continuum

This paper is articulated as follows: section 2 describes the case study from the point of view of its stakeholders. Section 3

quantifies the evolution of Moodle from two different points of view: at first, relatively to the active developers working on it; then, regarding the outcomes of the activity within the Moodle community. It also provides the basis for a framework of the transition of OSS projects to different stages of maturity. Section 4 finally concludes.

## II. MOODLE

This paper focuses on an extensive analysis (both qualitative and quantitative) of the business and development model of Moodle, a popular Open Source software for e-learning. Given its size, extensive development and user community, a more in-depth appreciation of Moodle, and how it achieved its status, is central to understanding Open Source software and its future among the software competitors.

Moodle’s development is centered around various actors:

1. **Moodle core developer:** Martin Dougiamas originally developed Moodle while working at his Ph.D. thesis in Curtin University of Technology, Australia. Now Moodle’s development is lead by Moodle Pty Ltd, a company he founded and leads.
2. **Commercial stakeholders and Moodle developers:** the entities that have an interest in the creation and support of Moodle:
  - *Moodle partners:* a number of organizations across the world who are directly contributing to the development of Moodle by way of funding or contributing their expertise. As we write (12/2009) there are some 50 partners, distributed across the Americas, Europe, Asia and Oceania. As yet there are no African partners.
  - *Commercial exponents,* **not** participating in the partnership, but working on the development of modules, plug-ins, themes and language packs.
  - *Moodle developers:* whilst Moodle’s development is lead by Martin Dougiamas through Moodle Pty Ltd, a large number of individuals have been contributing to the development of Moodle. Just over 200 developers have write access at this stage, but not all have been contributing into the source code.

Other developers do not have the right to publish their changes in the CVS tree (as quantified in the next sections). A Moodle partner or a commercial exponent may employ a number of developers.

- *Commercial exponents* focusing on installation, lightweight customization and support, but **not** providing custom development for Moodle.
3. **Moodle community:** this includes the large number of users of Moodle spread across 204 countries (September 2009). The community engages in Moodle's activities through on-line discussions in forums and in other specialised events. While the role of the community is important, Moodle is not led, as other projects, by the community.

#### A. MOODLE at the Open University

The Open University of the United Kingdom is a centrally funded higher education institution specializing in blended and distance learning, with an enviable reputation for its contributions to educational technologies. Recently, the Open University scored the highest student satisfaction rating in a National Student Satisfaction Survey covering England, Wales and Northern Ireland. Students are not required to satisfy academic entry requirements, which encourages participation from a diverse student body able to enrol and pursue the majority of awards and curricula.

Experiments with e-learning date from the mid-1980s and the spread of home computers. Computer conferencing was introduced to courses of 5000 students as early as 1989 followed by the first web sites in 1993. All these developments were bespoke and hence expensive to develop and maintain.

In November 2005, the Open University's Learning and Teaching Office (LTO) announced it was to commence a £5 million programme to “*build a comprehensive online student learning environment for the 21st century*”. Moodle is just one part of this student learning environment, but is the most visible from a student's perspective. The first courses were hosted in May 2006 at which time it was claimed to be the largest use of Moodle in the world.

### III. QUANTITATIVE STUDY

In these following sections it will be studied what happened in terms of effort provided and results achieved by the Moodle community. This will be done empirically, by analysing the public data pertaining the open development of Moodle. In terms of data sources, it has been established that different development practices have an influence on the best data source ([5], [17]), and that both the Configuration Management Systems (CMS) and the ChangeLog files offer more reliable information ([4], [11], [21]).

The steps to extract the information from the Moodle server, and to produce the results can be summarized in i) extraction of raw data, ii) filtering of the raw data, and iii) extraction of metrics. As part of these steps, Perl scripts were written to download, extract the activity logs, and parse the raw data contained in the CMS, and finally to extract pre-defined data fields.

#### A. Raw data extraction and filtering

The choice of the information sources was focused on the CMS commits of the system. The Moodle project maintains an own CMS server<sup>1</sup>, and the data contained spans some 9 years, between Nov 2001 and Aug 2009. Perl scripts were used to identify and extract every occurrence of the following items:

- *Committer:* contributor responsible for the commit;
- *Commit:* the detailed activity a committer was responsible for;
- *Date:* day, month and year of change.

The field *Commit type* includes: *File affected* (the name of the file created or directly modified by a change), and *Module* (the name of the subsystem a file belongs to). As mentioned above, two types of changes were considered in the present study: the creation of an element (a file or a module), and the modification of existing files or modules. After performing the extraction, we arranged the resulting data on a SQL table. It made up to some 72,000 entries, including new element creations and changes.

Apart from the basic information on the authorized committers to the Moodle CMS, several cases were identified were sporadic contributors (i.e., without a committer ID) submitted their code patches directly to the core Moodle developers. This additional information was also extracted, and some additional cleansing performed: for example, obvious variations of people ID's, in this case their email addresses, were mapped to one unique ID. Finally, the email address ID's relating to a known committer ID were converted into a single ID.

#### B. Metrics choice and description

The analysis of the Moodle system involved two types of metrics, used differently to discuss the research questions:

- **Input metrics (person/months):** the effort of developers was evaluated by counting the number of unique (or *distinct*, in a SQL-like terminology) developers during a specific interval of time. The chosen granularity of time was based on months: different approaches may be used, as on a weekly or on a daily basis, but it is believed that the month represented a larger grained unit of time to gather the number of active developers. This metrics was used to evaluate the first research question. For example, in Feb 2006, it was found that the core Moodle system had 22 distinct and active (i.e., committing to the CMS “moodle” trunk) developers. This accounts for 22 person/months.
- **Output metrics (modules/months):** the work produced was evaluated by counting the touches to files or modules during the same interval of time. Smaller-grained metrics, like lines of code, were not considered in this study: evaluating how many lines of code are produced by OSS developers could be subject to strong limitations<sup>2</sup>. In the following sections this metric will be used also as an indicator of parallel

<sup>1</sup> The web interface to the Moodle CVS is browsable at <http://cvs.moodle.org/>

development work performed in successful projects. As in the example above, it was detected that 594 distinct PHP files (contained in 163 distinct modules and sub-modules), were added or modified (but not deleted) in Feb 2006. This accounts for 163 modules/months.

In order to produce a realistic set of initial data, these metrics were monthly filtered, by excluding from the “Input” a specific committer ID (“moodlerobot”) responsible for automatic, uninteresting, commits (around 2,000 overall); and by excluding from the “Output” any activity concerning the “Attic” CMS location (which denotes deleted source material).

### C. Results

This section presents the main results obtained in the analysis of the commits and the committers' activity. The first set of results will summarize the analysis performed to measure the overall number of active developers, and the output produced. The second part will instead produce a description of the activities of one of the most productive commercial partner within the Moodle community<sup>3</sup> As an high-level objective, it was studied whether it was possible to trace the activity of commercial stakeholders in the development of this OSS system. In particular the results of Commercial OSS systems (Eclipse, as reported in [22]) are compared with those in Moodle as an example of Community OSS system.

	CMS entries	In CMS since	Distinct developers (overall)	KSLOCs (as of 09/09)	Files (as of 09/09)
moodle	77,100	11/01	120	293	1,587
patches	10,029	07/03	48	266	1,330
plugins	31,768	09/03	113	538	3,330
tools	1,981	02/04	23	38	104

Table 1: summary of characteristics of the studied directories in Moodle's CMS

#### 1) Input as Applied Effort

**Core Moodle.** The first attempt to collect data and results regarding the active developers of Moodle pertained only the “core” Moodle directory as found in the CMS server. Apart from this central component (which makes for the public releases), another folder (“contrib”) contains various code by contributors (organised in “plugins”, “patches” and “tools”), but is not wrapped in the official release.

As visible in Figure 2 (top), the first part of the evolution of the “core” system has the typical results of an early (or “cathedral” [5]) OSS project [8]: few contributors are visible in the first months, often the main Moodle developer (Martin

2 Lines of code produced are biased by the skills of the developer, the programming language and, in general, the context of the modifications.

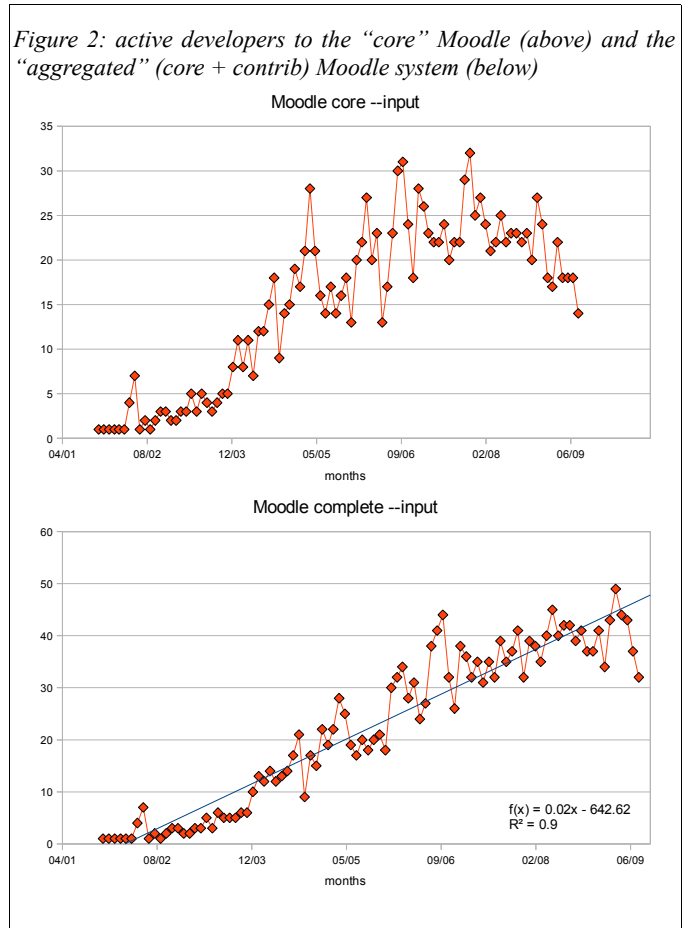
3 Catalyst, <http://www.catalyst.net.nz/>

Dougiamas), and few other sporadic contributors who are being active in a discontinuous way.

A second, more sustained period is then visible in the top part of Figure 2: the number of active developers follows a growing trend with peaks of over 30 developers a month contributing to the “core” Moodle. Albeit local maxima and minima appear frequently, the underlying increasing trend of the number of active developers reveals a “bazaar” phase [5], where the barriers to entry are more easily overcome by new contributors.

A third phase is finally present: during the last observed year, the number of active developers gradually started to peak off, and eventually to decrease. A further investigation into this revealed that one of the Moodle partners (Catalyst) gradually pulled its contributors from the development of the “core” Moodle.

**Aggregate Moodle.** The bottom part of Figure 2 is instead an aggregate view of all the contributors and developers working on both the “core” and the “contrib” components of Moodle. The more loose procedure to contributing to a non-core patch is visible in a continuously growing trend, that summed up to the “core” activity, still produces a growing



trend. The interpolation with a linear equation produces an adequate goodness-of-fit ( $R^2=0.9$ ).

As reported above, one of the most active committers among the Moodle partners (Catalyst) lately reduced the

amount of effort to be contributed to the community. Currently, a major factor that determines success or failure of an FLOSS initiative is whether its project management is in the hands of a commercial software corporation (the “Commercial OSS projects” of Figure 1, [6]). For example, the success of the Eclipse integrated development platform can be largely attributed to the project management of IBM. As demonstrated by the analysis of socio-technical relationships in the software change history of Eclipse [22], the upper part of the developer hierarchy is dominated by IBM staff. If this company decided to remove its support, the community around it would lose key developers and expertise.

In the Moodle analysis, the OSS community is in charge of most of the decision and development processes (as in “Community OSS projects”, [6]): if a commercial partner fails to provide its support, the development process does not come to a stop. This will be further validated in the Output section below.

### 2) Output as Produced Modules and Files

Similarly to what is done above, in this subsection it is studied the amount of output produced on a monthly basis. In particular the empirical analysis concerned the amount of “distinct” modifications to “unique” tuples (file, module): even if a specific file in a module was modified several times in a given month, the produced output would be only counted once. In the following example (Table 2), 5 modifications happened in 06/2002, 4 of which affecting the tuple (*index.php*, */cvsroot/moodle/moodle*), and one the tuple (*config-dist.php*, */cvsroot/moodle/moodle*). Since only the distinct tuples were considered in the analysis, from this example an output of 2 modifications is recorded for the month 06/2002. Below a summary of the findings regarding the output (both in the “Core” and in the “Aggregated” system) is given.

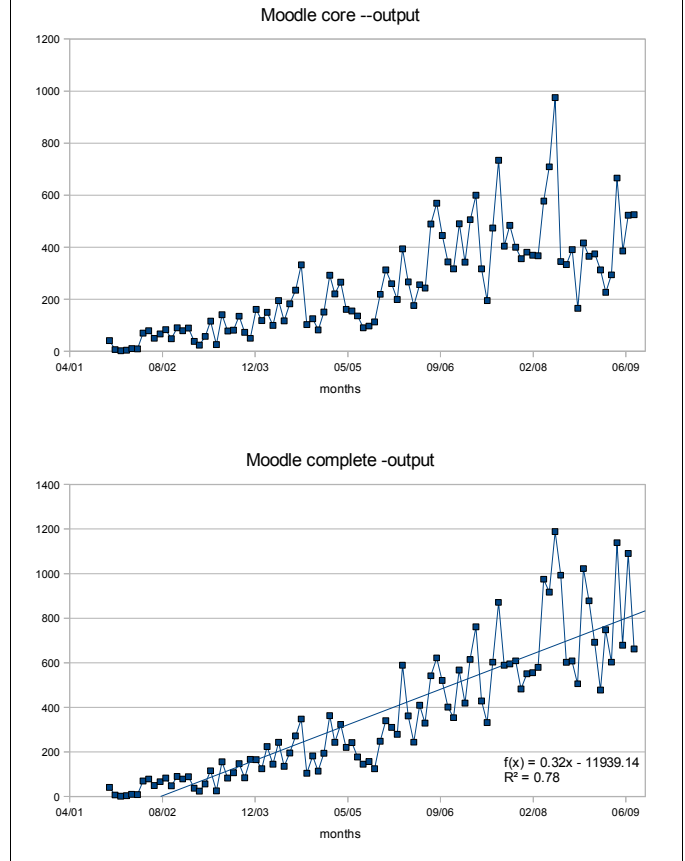
month	year	module	file	Developer ID
6	2002	/cvsroot/moodle/moodle	config-dist.php	martin
6	2002	/cvsroot/moodle/moodle	index.php	martin
6	2002	/cvsroot/moodle/moodle	index.php	martin
6	2002	/cvsroot/moodle/moodle	index.php	martin
6	2002	/cvsroot/moodle/moodle	index.php	martin

Table 2: excerpt from the CMS log of Moodle

**Core Moodle.** As distinguished above, it was first studied what visible output was obtained in the “core” Moodle system (first row of Table 1), by analyzing the additions and the modifications of PHP code in the “moodle” CMS directory alone. The results are shown in the top part of Figure 3: the initial phase mirrors the presence of just one developer, while the increasing amount of developers committing to the code becomes more and more evident with regular cycles of productivity, where the maximum peaks reflect the presence of public releases.

**Aggregate Moodle.** Repeating the process above, also the other contributed modules were analyzed with respect to the output obtained, and summarized in the bottom part of Figure 3. The overall distribution of changes throughout the Moodle evolution proceeds on a linear trend ( $R^2=0.78$ ): in recent months, the inflection of productivity in the “core” Moodle has been balanced by the late growth of contributions to the other parts. That reflects a more and more distributed participation to the Moodle development, and a low barrier to entry, but several of the proposed modules have not been selected for inclusion in the main Moodle system.

Figure 3: distinct tuples (file, module) as output produced in the “core” alone (above) and the overall Moodle (below)



### 3) The Catalyst involvement

As a further analysis, it was studied the specific involvement of Catalyst, a Moodle partner which has so far provided a large number of modifications to the core Moodle, by deploying several of its own developers who became active contributors within the community.

Since March 2004 Catalyst had from one developer up to a maximum of 6 developers (March 2005) working on Moodle. The profile of the contributed outputs is visible in Figure 4, and can be defined as a “seasonal” effort pattern, meaning a large contribution on a very specific time interval, and lower levels of effort before and after it. Also the modules developed by Catalyst are specifically targeted to a quite focused part of the core of Moodle: Figure 6 displays the distribution of effort along the modules, and it becomes evident how Catalyst wanted to be involved early on in the development of the

SCORM (Sharable Content Object Reference Model) collection of specifications.

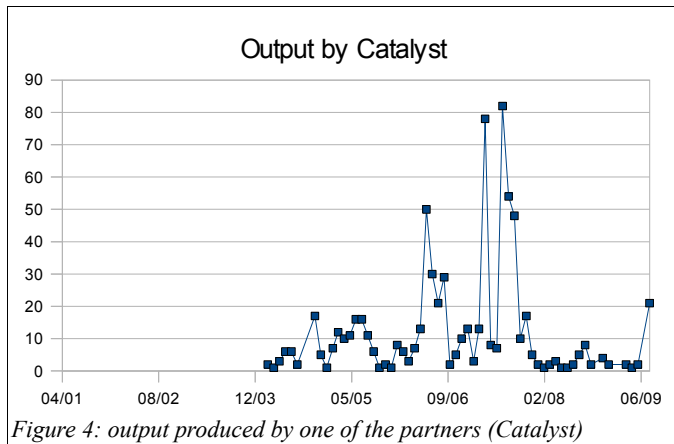


Figure 4: output produced by one of the partners (Catalyst)

The two Figures 4 and 6 show that the involvement of commercial entities follows the same principle of attracting individuals into an OSS community: they start to contribute to the periphery, then become more confident with the code, and have a peak of productivity, then leave [19].

The second observation shows that the “Community FLOSS” projects (from Figure 1) are not overly dependent on specific commercial stakeholders: the reduction of effort and output by Catalyst does not shrink the overall productivity. On the contrary, Commercial OSS projects (e.g., Eclipse), led and managed by specific companies (e.g. IBM) would probably collapse when the existing backing company decided to pull away its support.

modules worked on by Catalyst

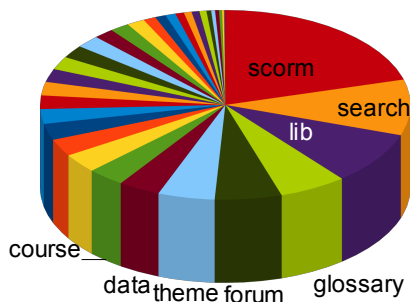


Figure 5: modules contributed by Catalyst

#### D. Discussion

The growth patterns of the Moodle system should not be considered as “exceptional”, but put in the context of a “framework of evolution” [2]. It is claimed that Moodle achieved two main transitions in its lifecycle: first, it overgrew the average projects in an OSS repository as SourceForge, so it needed to be pulled away from that repository; second, it achieved a transition from a “plain OSS” to a “Community OSS” project (as per the definitions of Figure 5). The two

transitions together have made this project a success, and they also put the foundations of “trust” in this system.

Also the Moodle project achieved such a transition: after 5 years of hosting on SourceForge, the Moodle team felt that the project had “outgrown it”<sup>4</sup>, so they moved to their dedicated web-server, and own CMS facility, in turn showing that the project has achieved a status similar to flagship OSS projects (like Apache or Linux and the such). This transition happened after a major interest of governmental and commercial stakeholders, which on the other hand helped Moodle to become a successful “Community OSS” project, with an increasing number of sponsors and partners that increased the perceived trust in it.

#### IV. CONCLUSIONS

This paper has proposed a quantitative study of the Moodle e-learning platform. This project started as a small project managed by only one developer on the SourceForge OSS portal, and is now used internationally, sponsored by several commercial partners and supported by even more commercial companies. Its usage and needs have grown to the point to require its own servers, and to gradually being pulled away from the SourceForge hosting.

This paper proposed the account of the largest e-learning provider in the UK, the Open University, which in 2005 migrated its technology to use the Moodle platform. In turn, this had the effect of becoming an active participant in the development process, and to increase the popularity and visibility of Moodle as a widely-spread solution for e-learning needs. The study of the evolution of Moodle (from its inception up to Sept 2009) visualised a recurring trend for FLOSS projects: a first stage of development where few developers are responsible for most of the activity, followed by a continuous growth that appears to be self-sustained. Even when one Moodle partner discontinued its contributions to the Moodle core, the development still appears to be not affected.

As a corollary, this project achieved a double transition: as mentioned in a previous research work, Moodle transitioned from an Open Forge (i.e. SourceForge) to a more defined, more successful status, as experienced by OSS projects transiting to more renowned and quality-stringent OSS portals. Secondly, starting from a community-driven OSS, Moodle has become a Community OSS project, where several commercial stakeholders start to act as sponsors of the project, increasing its visibility and establishing it as a de-facto standard in the domain.

#### V. REFERENCES

- [1] A. Baravalle, S. Chambers. Market Relations. Non-Market Relations and Free Software. *PsychNology Journal*, 5(3):299–309, 2007.
- [2] K. Beecher, A. Capiluppi, and C. Boldyreff. Identifying exogenous drivers and evolutionary stages in FLOSS projects. *Journal of Systems and Software* 82(5):739-750, 2009.
- [3] A. Bonaccorsi and C. Rossi. Altruistic individuals, selfish firms? The structure of motivation in open source software. *First Monday*, 1(9), January 2004.
- [4] A. Capiluppi. Models for the evolution of OS projects. In *Proc. of Intl Conference on Software Maintenance (ICSM 2003)*, pp.65-74, Amsterdam, Netherlands.

<sup>4</sup> “Moodle CVS has moved off Sourceforge”, news at <http://sourceforge.net/projects/moodle/>

- [5] A. Capiluppi, M. Michlmayr (2007). From the Cathedral to the Bazaar: An Empirical Study of the Lifecycle of Volunteer Community Projects. In: Feller, J., Fitzgerald, B., Scacchi, W., Sillitti, A. (Eds.), *Open Source Development, Adoption and Innovation*, pp.31–44
- [6] E. Capra, C. Francalanci, and F. Merlo. An empirical study on the relationship between software design quality, development effort and governance in open source projects. *IEEE Trans. Softw. Eng.*, 34(6):765–782, 2008.
- [7] J. Dalziel. Open standards versus open source in e-learning: The easy answer may not be the best answer. *Educause Quarterly*, 4:4–7, 2003.
- [8] Y. Fang, and D. Neufeld. Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, 25(4):9-50, 2009.
- [9] J. Feller, B. Fitzgerald, F. Hecker, S. Hissam, K. Lakhani, A. van der Hoek (eds.). *Characterizing the OSS process*, ACM, 2002.
- [10] M. Fischer, M. Pinzger, and H. Gall. Populating a release history database from version control and bug tracking systems. In *Proc of Intl Conference on Software Maintenance (ICSM 2003)*, pp.23-32, Amsterdam, Netherlands, 2003.
- [11] D. M. German. An Empirical Study of Fine-Grained Software Modifications. In *Proc. of Intl Conference on Software Maintenance (ICSM 2004)*, Chicago, US.
- [12] D. M. German, The gnome project: a case study of open source, global software development, *Software Process: Improvement and Practice* 8 (4): 201–215, 2004.
- [13] A. Hemetsberger, C. Reinhardt, Sharing and creating knowledge in open-source communities : The case of kde, in: *Proceedings of the Fifth European Conference on Organizational Knowledge, Learning and Capabilities (OKLC)*, Innsbruck University, 2004.
- [14] S. Koch, G. Schneider, Effort, cooperation and coordination in an open source software project: Gnome, *Information Systems Journal* 12 (1) (2002) 27–42.
- [15] M. Kuniavsky, S. Raghavan, Guidelines are a tool: building a design knowledge management system for programmers, in: *DUX '05: Proceedings of the 2005 conference on Designing for User eXperience*, AIGA: American Institute of Graphic Arts, New York, NY, USA, 2005.
- [16] P. B. de Laat. Governance of open source software: State of the art. *Journal of Management and Governance*, 11(2):115-117, 2007/
- [17] T. Mens, J. F. Ramil, and M. W. Godfrey. Analyzing the evolution of large-scale software: Guest editorial. *Journal of Software Maintenance and Evolution*, 16(6):363-365, 2004.
- [18] M. Michlmayr, A. Senyard. A statistical analysis of defects in Debian and strategies for improving quality in free software projects, in: J. Bitzer, P. J. H. Schrder (eds.), *The Economics of Open Source Software Development*, Elsevier, Amsterdam, The Netherlands, 2006.
- [19] G. Robles and J. M. González-Barahona. Contributor turnover in libre software projects. In E. Damiani, B. Fitzgerald, W. Scacchi, M. Scotto, and G. Succi, editors, *OSS*, volume 203 of *IFIP*, pages 273--286. Springer, 2006.
- [20] G. Robles, S. Duenas, J. M. González-Barahona. Corporate involvement of libre software: Study of presence in Debian code over time, in: J. Feller, B. Fitzgerald, W. Scacchi, A. Sillitti (eds.), *OSS*, vol. 234 of *IFIP*, Springer, 2007.
- [21] N. Smith, A. Capiluppi, and J. F. Ramil. Agent-based simulation of open source evolution. *Software Process: Improvement and Practice*, 11(4):423-434, 2006.
- [22] M. Wermelinger, Y. Yu, and M. Strohmaier. Using formal concept analysis to construct and visualise hierarchies of socio-technical relations. In *Proc. of the 31st International Conference on Software Engineering*, 18-24 May 2009, Vancouver, Canada.