



HAL
open science

Engaging without Over-Powering: A Case Study of a FLOSS Project

Andrea Capiluppi, Andres Baravalle, Nick W. Heap

► **To cite this version:**

Andrea Capiluppi, Andres Baravalle, Nick W. Heap. Engaging without Over-Powering: A Case Study of a FLOSS Project. 6th International IFIP WG 2.13 Conference on Open Source Systems,(OSS), May 2010, Notre Dame, United States. pp.29-41, 10.1007/978-3-642-13244-5_3 . hal-01056032

HAL Id: hal-01056032

<https://inria.hal.science/hal-01056032>

Submitted on 14 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Engaging without Over-powering: a Case Study of a FLOSS project

Andrea Capiluppi¹, Andres Baravalle¹, and Nick W. Heap²

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

2 The Open University
Walton Hall, Milton Keynes, UK
n.w.heap@open.ac.uk

Abstract. The role of Open Source Software (OSS) in the e-learning business has become more and more fundamental in the last 10 years, as long as corporate and government organizations have developed their educational and training programs based on OSS out-of-the-box tools. This paper qualitatively documents the decision of the largest UK e-learning provider, the Open University, to adopt the Moodle e-learning system, and how it has been successfully deployed in its site after a multi-million investment. A further quantitative study also provides evidence of how a commercial stakeholder has been engaged with, and produced outputs for, the Moodle community. Lessons learned from this experience by the stakeholders include the crucial factors of contributing to the OSS community, and adapting to an evolving technology. It also 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.

1 Introduction

In the first decade of the twenty-first century three factors have been pushing the “e-learning topic” under the spotlight: first, the recognition that it has become, together with the underlying technology, a recognized and sustainable industry. Secondly, the attempts to create Open Data Standards (ODS) for e-learning content, driven by specification organizations such as the IMS Global Learning Consortium, Aviation Industry CBT (Computer-Based Training) Committee (AICC), and Advanced Distributed Learning (ADL) network sponsored by the U.S. Office of the Secretary of Defense, and relevant committees of international standards bodies, such as the IEEE Learning Technology Standards Committee [7]. Finally, the much wider movement that advocates OSS and open data standards. OSS includes highly successful software such as the *Linux* operating system, the *Apache* web server and *OpenOffice.org*.

The increased importance of e-learning, the emergence of ODS for e-learning and the driving push from the Open Source community are creating a fertile environment where innovation can spread more efficiently.

E-learning platforms have been in use for a number of years, but the technological focus has been shifting: early adopters were focusing on e-delivery of teaching material, that is just allowing users to download teaching material from the web. At present, academic institutions are also focusing on other areas, and one of the most pressing issues is the packaging and distribution of e-learning resources. Academics, who are used to disseminate the findings of their research to the wider academic community, are less used to use, modify and redistribute teaching material. The complexity of e-learning software, poor interoperability and the elevate cost of commercial e-learning solutions play all a central role in this. OSS and ODS can help to address both interoperability and price, ensuring that teaching material can be exchanged and used more easily and with inferior economical costs.

This paper studies the evolution of the Moodle e-learning platform, and describes the process of its deployment in the the Open University, the largest on-line 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, since they are driven by the community, but they also often have several commercial stakeholders.

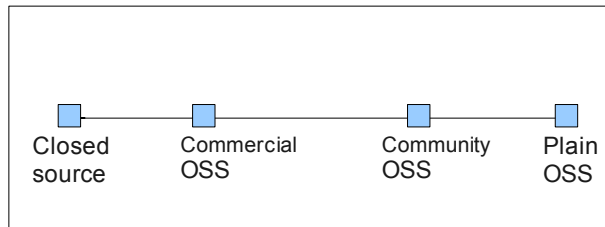


Figure 1: Software licensing continuum

This paper is articulated as follows: section 2 describes the case study from the point of view of its stakeholders. Section 3 focuses on one stakeholder (the largest provider of e-learning resources in UK), and illustrates the process of adopting this OSS solution, and the issues and benefits of doing so. Section 4 focuses on another

commercial stakeholder (Catalyst IT Ltd.) and quantifies its contributions to Moodle from the point of view of its developers deployed to Moodle. Section 5ss concludes.

2 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:
 - (a) 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.
 - (b) *Commercial exponents*, **not** participating in the partnership, but working on the development of modules, plug-ins, themes and language packs.
 - (c) *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.
 - (d) *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 (as of September 2009). The community engages in Moodle's activities though 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.

It is also important to note that Moodle has been building on existing technologies and research, from a community much wider than the sole Moodle community. For example, Moodle uses PHP and MySQL for server-side

development, incorporates previous works on data standards, and is supporting existing technologies as SCORM and LAMS to incorporate teaching objects.

2.1 Business model

Before proceeding, it's important to discuss and analyze the Moodle business model. While Open Source software is free to use, modify and redistribute, it does not mean that successful business modules cannot be created around Open Source software [1]. In the context of Moodle, a number of different strategies are currently pursued by Moodle Pty Ltd, the commercial stakeholders and the developers:

1. **Project lead:** Moodle Pty Ltd, and Moodle developers are in a privileged position to receive funding for additional features to be included in the system.
2. **Partnership synergies:** Moodle partners have a "privileged relationship" with Moodle Pty Ltd, For example, as we will see later, the development on Moodle itself was lead by a UK company rather than by Moodle Pty Ltd, Partners have a privileged access to local markets thanks to customer referrals, and at the same time provide an additional source for funding or resources for Moodle Pty Ltd.
3. **Peripheral development:** commercial exponents who do not have developer or partner roles typically work on the more peripheral areas of Moodle, which do not require changes in the core areas of the code base. As we will see in the next sections, certain stakeholders may find a number of strategies to be ineffective: trying to submit changes in core areas might prove difficult and thus can lead to expensive maintenance costs, as the commercial exponent would have to maintain its own fork of Moodle.

2.2 Commercial stakeholders and peripheral development

This subsection summarizes the experience of one of the commercial exponents, but not a Moodle member. Mediamasteri Group Ltd.¹ is a Finish leader in the area of virtual learning environments and as part of its business activities sells Moodle related services, such as maintenance, deployment, content production. In the past years Mediamasteri has been developing a variety of custom modules (about 15) and components, only some of which are used in the community version of Moodle. Between 2003 and 2006 a number of their modules have been approved and included in Moodle but from 2007 there has been a change in trends.

Although it has been supporting 5 to 10 developers working on Moodle in the previous years, now the investment on Moodle development has been slowing. Their modules are not making it in the official Moodle distribution and they find it hard to support them just with their own workforce. Similarly, their changes to core areas of

¹ <http://www.mediamasteri.com>

Moodle are also not making it to the official Moodle release, and they are now put in a situation where they have to support their own version of Moodle.

While active in Moodle development, Mediamasteri is not an official Moodle partner, nor any of its developers has official developer status in Moodle. The implication is that they are not involved in the planning phases, and they have a competitive disadvantage comparing to other companies.

Mediamasteri experience shows that, at least in case of Moodle, commercial partners are treated similarly to any other OSS contributor, and their code patches will go through the usual scrutiny from the community. Organizations (and individuals) like Mediamasteri who have a limited involvement (at least in terms of resources committed to the project) may find it difficult to modify core components.

3 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 established 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 enroll 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.

There are valuable lessons to be learned from the Open University's experience, such as how the institution arrived at its decision to use Moodle, what were the main issues in its planned development and deployment, and what benefits were gained from the early adoption of this OSS package.

3.1 Initial Selection

The selection phase for the core of the VLE platform commenced in 2003 and ran for almost two years. A complete review was undertaken of all existing support and delivery systems along with visits to other institutions to learn about their experiences with various platforms.

Early consideration was given to an in-house development that could tie together the mixed-bag of systems supporting registration, content delivery, and learning support, but it was quickly discounted as prohibitively expensive.

A range of proprietary solutions were also considered, but excluded because they offered limited customization and could not guarantee the scalability required; the Open University has some 150,000 students using its on-line systems. OSS solutions were reviewed and initially rejected because of concerns about the high level of risk and the lack of a viable partner.

Having eliminated all the options the review team went back to investigate a combination of in-house development coupled with an OSS learning platform. By September 2005 the business case was completed and the formal decision to adopt Moodle was announced in November 2005.

The substantive development phase has now drawn to a close and all courses migrated to the new platform. Although some work remains, the time is right to reflect on what has been achieved and what lessons have been learnt. As a consequence a small number of interviews have been undertaken with development staff including the Director of the Learning Innovation Office, various Project Leads, and individual programmers.

The selection phase had established some 23 areas of development work that would be required to add or enhance Moodle features – as they existed in 2005. Of these the following were regarded as potential 'show stoppers':

1. **Existing user model:** the Open University's student support model and administration system requires a hierarchy of user roles (and associated permissions) to support the various combinations of full-time and part-time teaching staff, editorial, production, and technical, and the Help-Desk. A typical course may have as many as 30 user categories whereas Moodle supported just three roles.
2. **Limited database support:** Moodle offered no support for either Microsoft's or Oracle's RDBMS, which were the database servers in use in the university. Furthermore, a database abstraction layer was missing.
3. **Grade-book feature:** a new facility to permit students and teaching staff to review assignment grades.
4. **Data entry forms:** inconsistent coding of data and text entry forms contributing to poor accessibility and difficult maintenance.

One of the greatest challenges for the Open University was to balance the benefits of the Moodle solution against the costs of the enhancements to ensure fitness for purpose. The benefits of adopting an off-the-shelf solution would quickly disappear if too many in-house changes were implemented prior to deployment.

Discussions within the Moodle community concluded that items 2, 3 and 4 would enhance the Moodle core and so the Open University agreed to fund the development costs by contracting out the work to Moodle Pty. As a result of this effort, a database abstraction layer was created (named XMLDB) and the API improved.

On the other hand, the 'user model' changes were regarded as controversial: many in the community considered the changes unnecessary, whilst others were concerned about the potential impact on performance: as a result this effort was undertaken by the Open University.

3.2 Role of the Open University as Community Contributor

The challenges faced by the Open University are common to any organisation that contemplates the introduction of an OSS solution into its core business functions. However, the attitude of many is to adopt OSS packages behind the scenes, possibly adapting it to fit some niche requirement, but in general to avoid full participation in the community [20]. They may partake of the community support, which in some cases is extremely fast, or use the OSS brand to distinguish themselves from their competitors. At worst they may be viewed as exploiting the computing skills of the community. What is clear is that their motivations are very different from the individual developers who decide to invest their efforts into an OSS community [3].

Perhaps it is not surprising that organizations are cautious when it comes to community participation, for as the Open University had to learn, there are significant challenges to becoming a full and active member of an OSS community. In this regard the Open University faced two major challenges. The first was to understand the philosophy of an OSS community, how it operates, how consensus is achieved, and the pace at which change occurs. The second was to come to terms with the underlying technology of Moodle.

The concept of “contributing to an OSS community” was new to the Open University. As a well known national organization, it was more familiar with a commercial procurement model of purchasing, based on requirements and specifications, with fixed delivery dates and penalties for non-compliance.

The second challenge was that the Open University's in-house team had very limited experience of developing with Moodle's programming language, PHP. Since PHP is an Object-Oriented (OO) language, the Open University developers erroneously assumed that they could migrate their OOs skills and practices directly to the Moodle community. Instead, the developers found that their solutions either didn't work properly, or impacted Moodle's performance.

The following issues were reported by interviewees when the Open University attempted to contribute code, or proposals for enhancements, to the Moodle community:

- **Coding standards:** early versions of Moodle showed wide ranges of coding skills and practice. In order to ameliorate this, recently the overarching Moodle Pty company has created, and is enforcing, coding and documentation standards. What was also done in this respect was to bring in more stringent review procedures: they are applied both to proposed changes to Moodle and during the implementation phase.

- **Rejection of contributions:** some of the refinements proposed, or completed, by the Open University were rejected after discussion for the core trunk of the Moodle project. For example, the Open University's Wiki development was rejected by the community, even though the change itself was later supported by Moodle Pty, due to cleaner code, and a better usability.
- **Slow uptake of contributions:** although the Blogs and Wiki developments, undertaken by the Open University, have been contributed back to the community (in the “plugins” section), the uptake from the community was low.
- **The contribution process:** in general, it was felt that contributing to the Moodle community is often hard work. Proposals of development must be developed first, and time has to be allowed for their public reviews. In some cases, these reviews may even highlight secondary changes, that eventually increase the costs of the proposed development. Only occasionally it was found that the requests from the Open University and Community coincide, and in those cases the contribution process was facilitated.
- **Support to the contribution:** within the Moodle community it is accepted that the contributions require support during the early stages of testing and deployment. It is also a shared expectation within the community that the original contributor will support his/her changes. Even in this case, this cost may not be that significant, as the Open University would have to test and maintain for its own code-base anyway.

Apart from these aspects, the Open University has been greatly benefiting from Moodle, as summarised in the following points:

- **No license fees:** while the Open University has invested considerable amounts of money, they are now free from license fees.
- **Maintenance:** the code adopted for core is maintained by Moodle Pty so will be retained in future releases
- **No vendor lock-in:** the Open University has already experienced the problem of vendor lock-ins in the past. With thousands of modules running, vendor lock-ins can be problematic and migration to a different technology prohibitively expensive.

3.3 The Open University and the Open Source community

The Open University avoided to create its own fork of Moodle in order to maximise the interaction with the community. When new versions are released, all that is necessary is to replace those modules that provide connectivity with other systems, but the process has been semi-automated. A recent update of the entire system required only half a day.

Early on in its Moodle development process, the Open University has been releasing too many plug-ins requiring changes to the core, but these were not widely

adopted. Familiarity with Moodle's code has reduced this. More recent developments by the Open University have been contributed back and have been better received, for example a new “Session data storage” feature, by using a file server, and fine tuning.

Moodle 2.0 will bring a number of changes, and will reflect the influence of the Open University. It will also reduce the number of Open University changes to the standard core, but is likely to increase problems for the tailored plug-ins that the Open University has been developing.

The Open University has provided a number of benefits for the community:

- Bug reports, code reviews, feed-back.
- Financial help: funding for improvements in a number of critical areas, especially related to scalability and performance.
- Valuable proof of concept: the Open University's Moodle installation is used in a high availability environment and it has to sustain high levels of load. This in turn has proved that Moodle is a commercially viable product for even the most challenging environments.
- An improved image: the Moodle community can count-in a prestigious university. This has clearly benefits for companies providing Moodle consultancy.

On the other hand, the presence of the Open University as a stakeholder within Moodle has also produced some disadvantages:

- Scheduling of development work and release dates: the Open University has a long lead-time for course development and production, typically 2-3 years, so needs to know when features will be available so that they can be incorporated into new courses.
- The Open University might be, at same stage, an intrusive guest. With over 450,000 users of their courses (between OpenLearn and the paid-for courses) and nearly 3,600 active modules, they can have an important influence in the development of the software. Smaller Moodle installations have typically different requirements, priorities, complexities.

4 THE CATALYST INVOLVEMENT

In the previous sections, a report of what Moodle achieved in terms of popularity, the tiers of its development, and the involvement of the largest e-learning institute in UK was documented. As a further analysis, it was studied the specific involvement of *Catalyst IT Ltd* (“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.

The analysis of Catalyst's involvement was achieved 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 regarding Catalyst were 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.

4.1 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 server², 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.

4.2 Metrics choice and description

The analysis of the Moodle system involved the analysis of *input metrics*: 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 (i.e., man-month).

² The web interface to the Moodle CVS is browsable at <http://cvs.moodle.org/>

4.3 Results

This section presents the main results obtained in the analysis of the Catalyst involvement in the Moodle development. As an high-level objective, it was studied whether it was possible to trace the activity of this commercial stakeholders: in particular, the results of *Commercial OSS* systems (e.g., Eclipse, as reported in [22]) should be compared with Moodle as an example of *Community OSS* system.

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 2, 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 3 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.

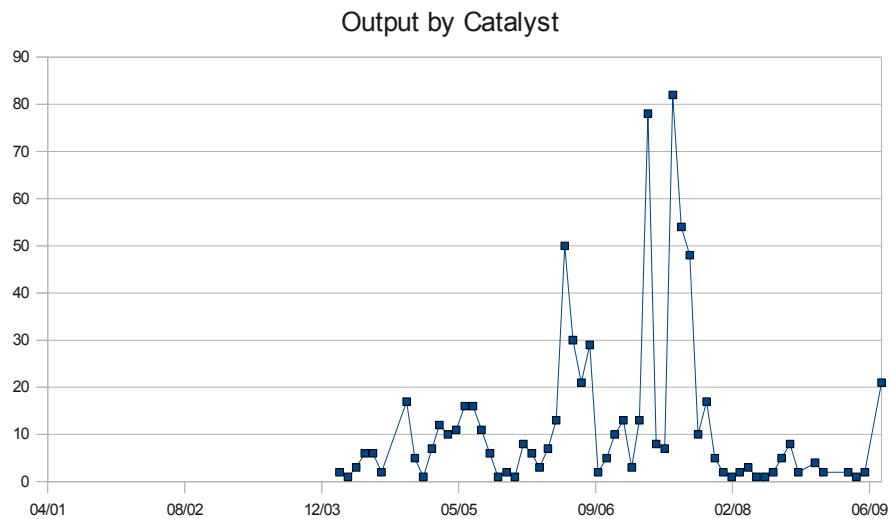


Figure 2: output produced by one of the partners (Catalyst)

Figure 2 and Figure 3 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 OSS projects (from Figure 1) are not overly dependent on specific companies: 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 company decided to pull away its support.

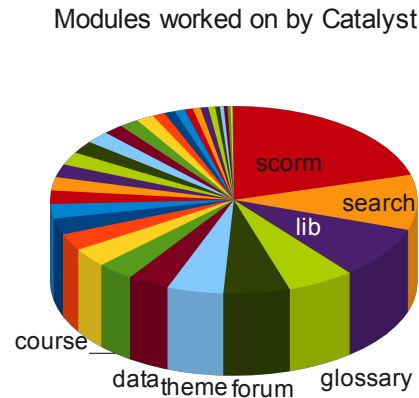


Figure 3: modules contributed by Catalyst

5 CONCLUSIONS

This paper has proposed a mixed qualitative and quantitative study in order to study 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 was essentially two-fold: at first, it 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 second strand of research quantitatively studied the quantitative involvement of a Moodle partner (Catalyst IT Ltd) during its evolution, and recognised an established trend for OSS contributions: a first stage of development where a limited output is contributed, then a peak of contributions, finally the abandonment of the commitment. The overall development of Moodle still appears not to be affected, even when Catalyst discontinued its contributions to the Moodle core.

As a corollary, this project achieved a double transition: as mentioned in a previous research work [2], Moodle transitioned from an Open Forge (*i.e.*, SourceForge) to a more defined, more successful status, as experienced by OSS projects transitioning to more renowned and quality-stringent OSS portals. Secondly, starting from a “pure” OSS project, 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.

6 ACKNOWLEDGEMENTS

The authors wish to thank the Open University staff who were interviewed for this paper. Mediamasteri for their feedback and the clarification of their involvement in the community development, and the Remote-Learner Canada, who provided feedback on their involvement within Moodle. Finally we wish to extend our gratitude to the anonymous reviewers of the paper, who contributed valuable feedback, apart from a more apt title for this paper.

7 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.
- [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., Silitti, 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.