Collaborative Environments in Software Engineering Teaching: A FLOSS Approach

Sara Fernandes[1]and Luis Soares Barbosa[2]
[1,2]HASLAB INESC Tec, University of Minho, Portugal
[2]United Nations University, Campus de Couros, Portugal
sara.s.fernandes@inesctec.pt
lsb@di.uminho.pt

Abstract: Open development has emerged as a method for creating versatile and complex products through free collaboration of individuals. This free collaboration gathers globally distributed teams. Similarly, it is common today to view businesses and other human organisations as ecosystems, where several participating companies and organisations co-operate and compete together. As an example, Free/Libre Open Source Software (FLOSS) development is one area where community driven development provides a plausible platform for both development of products and establishing a software ecosystem where a set of businesses contribute their own innovations. Equally, open and informal learning environments and open innovation platforms are also gaining ground. While such initiatives are not limited to any specific area, they typically offer a technological, legal, social, and economic framework for development, relying always on people as open development would not exist without the active participation of them. This paper explores the participation of master students in FLOSS projects, while merging two different settings of learning: formal and open/informal education.

Keywords: free/libre open source software, software engineering education, participatory learning, community driven development, collaboration, formal learning, informal learning, open learning

1. Introduction

Open development has emerged as a method for creating versatile and complex products through free collaboration of individuals, and gathering distributed teams. Initiatives such as the Open Government Partnership (Open Government Partnership , 2008), the Open data (Open Knowledge International , 2009), or Free/Libre Open Source Software development (FLOSS) (Stallman, 2015) are becoming more and more prevalent.

FLOSS development is one example where community driven development provides a plausible platform for both development of products and establishing a software ecosystem where a set of businesses contribute their own innovations. While such initiatives are not limited to any specific area, they typically offer a technological, legal, social, and economic framework for development, relying always on people as open development would not exist without the active participation of them. There are several examples of FLOSS development with business impact that are very well know, and are used on a daily base worldwide. Among those examples we can name Wikipedia (Wikipedia Foundation, 2001), Firefox (Mozilla, 2002), or Thunderbird (Mozilla, 2004).

Current trends in education point out that learning results from participation in social interactions and in culturally organized activities with others (Palincsar, 2013). This shift of perspective raises a number of questions on the organization of the educational process, namely on its dynamics to provide suitable support with different degrees of formality. E-learning systems and e-learning supported infrastructures are certainly part of this debate. It has been pointed out that, “in the last 20 years, e-learning grew from a unique college experiment to a full category of higher education. In 2010, there were more people enrolled in online classes than the entire population of Wisconsin” (elearners, n/a). It is therefore legitimate to think of a fully personalized education system designed around needs, interests and aspirations of each learner. Moreover, with the emergence of Web 2.0, conventional e-learning systems, based on instructional packets and cumulative assignments, gives the stage to a different reality which promotes the concept of social learning through the use of social software tools, such as blogs, wikis, forums, etc. As a result, learning in a broad and heterogeneous perspective, occurs at a societal level through the development of complex interactions between peers. As such, living and working in a modern society requires observation, awareness and understanding how the social, economic, cultural, etc. environment is changing around us and how we tend to react to such changes. It requires willingness and ability to reflect upon, learn and eventually adapt to change (Free Management Library, 2016). This, in turn, is increasingly associated with continued learning. Continued learning is not about mere acquisition of content knowledge through formal courses. Instead, it is about building cross-cutting competencies and skills in reflection and inquiry, so that one’s life and work experience becomes a personal learning lab. Continued

learning involves developing knowledge, skills and capabilities in conceptualizing and taking responsibility for one's own learning process; adapting the learning process to a variety of living and working situations; continually observing and analyzing one's own experience to draw conclusions and insights and to inform personal decisions; thinking holistically about our presence in a larger "system" and our experience as a reflection of this system; among others (European Commission, 2007). This new learning perspective is in contrast with a number of fundamental assumptions which have historically underpinned the organization of education: 1) expertise and knowledge resides only within the walls of the educational institution; 2) "learning" and "schooling" are different words for the same thing; 3) the most "equitable" educational systems are those which offer a "one-size-fits-all" approach, and 4) the easiest and most cost-effective approach to organizing learning is within the walls of the school.

This paper aims at making a concrete contribution to re-thinking educational practices in computer-oriented environments. Far away from the "school as a factory" metaphor, we envisage learning approaches for software engineering courses that, in sharp contrast with formal institutions and curricula, promotes the usage of informal learning mechanisms, promoting the "learning by doing" approach and that, ultimately, prepares students for the future. Therefore, this paper reports on exploring, through a concrete initiative, the participation of master students in FLOSS projects, while merging two different settings of learning: formal and open/informal education.

The rest of the paper is structured as follows. Section 2 presents the background. Section 3 describes the research methodology used to run the pilot project on collaborative education with master students. Section 4 presents the project and its finding. Section 5 concludes.

2. Background

Open development has emerged as a method for creating versatile and complex products through free collaboration of individuals. This free collaboration gathers globally distributed teams. As an example, Free/Libre Open Source Software (FLOSS) development is an area in which community driven development provides a plausible platform for both development of products and establishing a software ecosystem where a set of businesses contribute their own innovations. Equally, open and informal learning environments and open innovation platforms are also gaining ground. While such initiatives are not limited to any specific area, they typically offer a technological, legal, social, and economic framework for development, relying always on people, their skills, capacities and participation.

FLOSS communities consist of heterogeneous groups of independent volunteers, who interact even if driven by different motivations (Cerone, 2010). In such communities, volunteers collaborate, share their knowledge, by not only teaching but also by learning.

Learning, on the other hand, can be defined as a "persisting change in human performance or performance potential which must come about as a result of the learner's experience and interaction with the world" (Driscoll, 2005). It can be formal, i.e., institutionally framed and hierarchically structured, or informal.

Informal learning is a life-long process in which an individual acquires knowledge, attitudes, values and skills while performing daily activity within various contexts. From Jay Cross' perspective, "people informally acquire much of the knowledge they use in their practice. Through the observation of others, by trial and error, and simply working side by side with more experienced people". In his opinion, "formal education contributes only about 10% to 20% of what a person learns in a professional context" (Cross, 2006). In both settings, the qualifier collaborative refers to sets of activities involving a group of people learning or trying to learn something together. The term can be defined more broadly as collaborative teaching and learning (Elizabeth F. Barkley, 2004), as both activities occur together. Unlike individual learning, collaborative teaching and learning capitalizes on students' resources and skills. For example, individuals learn from each other and teach to each other by enquiring, debating, cross-assessing ideas between members and mutually monitoring work progress. Collaborative teaching and learning encourages knowledge construction, skill development and a deeper understanding by actively engaging students in the learning process.

Any collaborative teaching and learning agenda, as well as any technology enhanced learning framework, will be based on assumptions, implicit or explicit, concerning what it means to learn in collaborative settings.

The theoretical lens that guides our study is constructivist epistemology (Piaget, 1976), which emphasizes the agency of the learner in the learning process. Learning can only happen through the learner's efforts at producing meaning (in a broad sense, making sense of the world), although a mentor might help in providing challenging experiences in order to accelerate the process of change (Suthers, 2006).

3. Research methodology / case study

Our research is based on a pilot project in teaching and learning software engineering that has been carried out for 2 years. The first findings were reported in the following publications: "Integrating Formal and Informal Learning through a FLOSS-Based Innovative Approach Background: Learning as a Process" (Fernandes, 2013), "FLOSS Communities as Learning Networks" (Fernandes, 2013), and "A pilot project on non conventional learning" (Fernandes, 2013).

The research follows a participatory action research approach analysed through the construction of a case study. The pilot project involves students, who act both as participants - involving themselves in the activities carried out within the project, and as observers - reflecting about their own practices, behaviours and achievements exhibited and gained through their participation in the project. They are part of a class of pre-service teachers, i.e. students in the last year of a MSc course whose completion will entitle them to teach Informatics at secondary school level. As such, they seem highly motivated to analyse new learning experiences and even test them in their own classes. By definition, participatory action research aims to understand the "world" by trying to change it, collaboratively and reflectively. Rather than a strict method, it is an approach to what research actually means in Social Sciences and Education.

The pilot project aims at teaching students' software engineering skills through their involvement in a FLOSS project, using the open and democratic style typical of FLOSS communities. Students are proposed a list of FLOSS projects among which they can choose one to get involved in, but they are also free to choose a project not in the original list. How students get together in small groups (up to 3 elements) and which role each student and/or small group will play within the project are also free choices. Within each group, leadership may spontaneously emerge and either have an official recognition or just appear as part of the interaction activities. Along the case study, data is collected through a combination of direct observation during a 2 hours' weekly meeting, unstructured interviews and independent analysis of the work provided in the community by the instructor. Interpretation of direct observation allows us to gather information about the learning and communication skills of the students, their interaction and collaboration modalities, and how roles and leadership emerge from the collaborative process. Unstructured interviews provide a more complete picture of students' behaviour by investigating actions and tasks that are not directly observable and fostering the externalization of motivations and expectations. The analysis of each project as a participant by the instructor allows to better understand the dynamics of all participants in a certain project. All data collected is stored in the project collaborative platform hosted by Moodle, and maintained by the instructor. The weekly meeting of all groups with a member of the research team (instructor) allows a live interaction and smooths some difficulties in the project development, namely at the technical level.

4. Project launch and findings

This paper explores the participation of master students in FLOSS projects, while merging two different settings of learning: formal and open/informal education. To this aim we ran a project with master students, comprised by pre-service teachers. The project had the participation of 16 students during the first year and 21 students during the second year.

4.1 Project launch

The project had the duration of 1 semester in both years of the experiment.

In the first year the pilot was planned as followed: during an introductory meeting the project was presented, and each student filled a questionnaire aiming at 1) understanding the academic and professional background of each student and 2) if students were familiar with FLOSS and FLOSS projects, both as users and contributors. After presenting the objectives of the project and the questionnaire was filled, and students, between them, gathered as groups of no more than 3 elements. Since the class had been working together for more than a year – the first year of the master degree – it was easy for the groups to be gathered.

During the first meeting, students were introduced to the platform they were going to work with, Moodle. Using Moodle, students downloaded the list of FLOSS projects recommended for the project. Since it was a group work, each group had one week to study each project and choose the one that was more suitable for them. It was also allowed for each group to choose a project that was available on GitHub or Source Forge. Added to the list, the research team also decided roles to be performed in each project. The roles could be: analyst, expected to document the functionalities of the software; programmer, to develop and integrate code; and tester, to test the software developed.

Once each group decided on the project, they start the technical work, such as installing all the required tools, and planning their activities. All tasks were reported on a weekly basis on the platform highlighting all the achievements and difficulties faced during each week. Finally, the final part of the project required 1) the submission of the work done to the relevant FLOSS community of each project; 2) the presentation to the whole class and instructors the work done during the semester.

During the second year of the experiment, motivated by a students' request, checkpoints were introduced so that each group could present on-going work at different moments. This made easier for each group to stress its concerns and achievements; additionally, it helped instructors to follow the work.

4.2 Findings

From the questionnaire we found out that the students involved had, on average, a modest background when compared to typical programming skills of members of FLOSS communities. All of them, however, were aware of the FLOSS phenomena and knew (or, at least, have heard about) a number of open source projects: the majority mentioned Linux, Open Office, and/ or Mozilla. This first activity allowed us to define the profile of the students: modest background in software development, very interested to learn new teaching methods and curious about FLOSS development. This aspect was later confirmed by the roles they selected; typically students with less skills on programming decided to perform roles such as analysts.

All students –both from the first and second year - were involved in the project for nearly 5 months. In both experiments their global attitude was pro-active, namely in dealing with difficulties in establishing a connection with the chosen FLOSS community. It was clear that the main challenge of such project was to ensure an effective interaction with the relevant FLOSS community – if in some cases it was a smooth activity, in others students faced many difficulties and ultimately had to change to a different project. Another finding of the participation in FLOSS projects and their communities was that, despite some communities welcomed the participation of students, even if in some cases, the community was very slow to answer; in others the community had some difficulty in understanding what the group was proposing to do. Typically, the interconnection with the communities, usually through a leading person of the community, was set in a mutual understanding basis, and within 30 to 40 days. This number seems too high with respect to our expectations of a live interaction with "live" communities and it is clearly a factor that needs to be taken into consideration when planning similar projects.

The second problem faced by the groups has normally to do with project management. Once students disclosed they were going to contribute to the project and study the community for only 5 months, the project management of most projects was not so welcoming of the idea, as they normally do not enjoy short term contributions. As such, for the second year, the research unit requested students not to disclaim immediately the time they were going to contribute to the project and study the community.

Interestingly, during the second year and because some projects were the same as the previous year, some group integrations and dynamics went smoothly, naturally taking advantage of the previous acquaintance of the 1st year of the project.

As far as the group task division was concerned, and differently of what the research team was expecting, small groups were quickly able to specialize each member in a particular task. In groups of 3 students, typically one was designated to lead the interaction with the community, another assigned the technical task of downloading, installing and configuring the software (namely in the beta- version in which the community was active) and finally, other became in charge of documenting the whole process.

The daily supervision of the project platform (based on Moodle) allows us to say that all groups were active in using discussion groups, chats, emails, and forums to exchange ideas, doubts or achievements, using similar channels that are actually using in each FLOSS community. They even made a number of suggestions to the research team to improve the collaborative platform. During the first year, and by students' initiative, an informal workshop, in which each group presented their own experience, was planned as a project checkpoint. This initiative was then formally adopted for the following year.

From the weekly meetings and the reports, it was interesting to see the impact that the "learning by doing" brings to someone's learning process. If depending on a certain community participating in a FLOSS project may bring some delays – either contributors are not so interested in the project or they don't have too much time for tutoring – also pushes students to be more pro-active in their learning process. Ultimately, this was necessary as the work performed was used for course assessment purposes.

Another interesting finding of the 2 years' project concerns how their academic and professional background limited the choice of tasks. Students with a more multi-disciplinary background (even if with a big component of technology) tend to choose tasks such as analyst rather than developer or code analyst. The fact that most of the students already have a background of working, and some where already above 35 years old, we found that there are not so willing to "think out of the box" and tended to be more conservative. Also interestingly is that when pushed, for example by someone the group or an interaction with the relevant FLOSS community, students showed an effort to push their own boundaries. Interestingly, another fining of the project is that FLOSS project contributors are not willing to teach but rather to discuss ideas and problems and show that they know how to fix problems.

Overall the participation in FLOSS project was a positive experience for the students. From both years of the experiment we can report that there were groups that: 1) struggled with the community as they did not reply or were taking too much time to reply; 2) we very welcomed by the chosen FLOSS community; 3) the community only took them seriously after the first commit they actually made; 4) none of the groups had difficulties with their group partners (at least nothing was reported); 5) once the community saw they were seriously in their task, they pushed the group to do more tasks; 6) almost all students felt their work was valid and appreciated; 7) students with more difficulties in software engineering were able to identify exactly their weakness but also where and how they could contribute best.

Finally, and during the final presentations, it was interesting to see how students were proud of the work they did, saying that it was very interesting to see people from all over the world praising their work and motivating them to continue their contribution after the experiment was finished.

5. Conclusions

Current thinking on education, recognizes that fundamentally learning and development result from participation of students in social interactions and culturally-rich activities with other students, and that IT has the potential to mediate learning experiences and to support networked collaborative learning. Such principle drove our research to whether FLOSS-driven projects do provide an interesting setting to exercise "learning-by-doing" and, in general, autonomous and proactive approaches to learning. From the two years experiment we conclude that participating in FLOSS projects can be really helpful as students have the opportunity to contribute to software development in real time and with real impact, and move inside a challenging and open (often too open …) technological and social framework.

From our analysis we can conclude that participating in FLOSS projects have a positive impact and that its inclusion in Software Engineering curricula as a complement to the formal education should be seriously considered. A number of aspects, however, should be taken into account: 1) students participating in such projects need to have a good background in software development and programming skills, and 2) FLOSS communities should be regarded only as supporters as, in general, they are not willing to spend much time to teach but rather to assist in problems that may occur.

From the 2 years' project, we confirmed our initial intuitions: Software Engineering curricula should be revised to include informal learning experiences. Participation in FLOSS projects, with suitable mentoring, may be one such possibility. Indeed, as more general claims, we would like to put on the table, for further analysis, the

following statements 1) expertise and knowledge do not reside only within the walls of the educational institution; 2) the most "equitable" educational systems are not those which offer a "one-size-fits-all" approach, and 3) the easiest and most cost-effective approach to organizing learning is not only within the walls of the school.

Acknowledgements

This work was supported by the Programa Operacional da Região Norte, NORTE2020, in the context of project NORTE-01-0145-FEDER-000037. The second author is further supported by FCT under grant SFRH/BSAB/113890/2015.

References

Barkley, E. ,. C. K. a. M. C., 2004. Collaborative teaching and learning techniques: A handbook for college faculty. Jossey-Bass. 1st Edition ed. .: Jossey-Bass.

Cerone, A. a. S. S., 2010. Using Free/Libre Open Source Software Projects as E-learning Tools. Pisa, Italy, ECEASST, p. 17.

Cross, J., 2006. Informal Learning: Rediscovering the Natural Pathways That Inspire Innovation and Performance. 1st Edition ed. .: Pfeiffer.

Dillon, T., 2006. The potential of open source approaches for education. [Online] Available at: http://archive.futurelab.org.uk/resources/publications-reports-articles/opening- education-reports/Opening-Education-Report200 c [retrieved 3 May 2012].

Driscoll, M. P., 2005. Psychology of Learning Instruction. 3rd Edition ed. Florida: Pearson Education, Inc..

elearners, n/a. 3 Reasons Why E-Learning Is Bigger and Better than Ever. [Online] Available at: http://www.elearners.com/online-education-resources/online-learning/3-reasons-why-e-learning-is-bigger-and-better-than-ever/ [retrieved 23 November 2012].

European Commission, 2007. Adult learning: It is never too late to learn. [Online] Available at: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=URISERV:c11097 [retrieved 1 July 2016].

Fernandes, S. C. A. a. B. L., 2013. A pilot project in non-conventional learning. Kent, UK., ACM, p. 346.

Fernandes, S. C. A. a. B. L., 2013. FLOSS Communities as Learning Networks. International Journal of Information and Education Technology , 3(2), pp. 278-281.

Fernandes, S. M. M. H. C. A. a. B. L., 2013. Integrating Formal and Informal Learning through a FLOSS-Based Innovative Approach. New Zeland, Springer-Verlag Berlin Heidelberg, pp. 208-214.

Free Management Library, 2016. Ways to Look at Training and Development Processes: Informal/Formal and Self-Directed/Other-Directed.. [Online] Available at: http://managementhelp.org/training/methods/formal-and-informal-methods.htm [retrieved 6 July 2016].

Mozilla, 2002. Firefox. [Online] Available at: https://www.mozilla.org/en-US/firefox/products/ [retrieved 6 January 2016].

Mozilla, 2004. Thunderbird. [Online] Available at: https://www.mozilla.org/pt-PT/thunderbird/ [retrieved 6 January 2016].

Open Government Partnership, 2011. Open Government partnership. [Online] Available at: http://www.opengovpartnership.org [retrieved 1 June 2016].

Open Knowledge International , 2009. Open Data. [Online] Available at: http://opendatahandbook.org/guide/en/what-is-open-data/ [retrieved 1 June 2016].

Piaget, J., 1976. The grasp of consciousness: Action and concept in the young child. 1st Edition ed. Cambridge, MA: Harvard University Press.

Scott, S. a. P. A., 2013. Sociocultural Theory. [Online] Available at: http://www.education.com/reference/article/sociocultural-theory/ [retrieved 3 March 2016].

Stallman, R., 2015. FLOSS and FOSS. [Online] Available at: https://www.gnu.org/philosophy/floss-and-foss.en.html [retrieved 12 January 2016].

Suthers, D., 2006. Technology affordances for intersubjective meaning making: A research agenda for CSCL. International Journal of Computer Supported Collaborative Learning, 1(3), pp. 315-337.

Wikipedia Foundation, 2001. Wikipedia. [Online] Available at: https://en.wikipedia.org/wiki/Wikipedia:About [retrieved 14 June 2016].