

# Integrating formal and informal learning through a FLOSS-based innovative approach

Sara Fernandes<sup>1,2</sup>, Maria Helena Martinho<sup>3</sup>, Antonio Cerone<sup>1</sup>, Luis Soares Barbosa<sup>2</sup>

<sup>1</sup>United Nations University – International Institute for Software Technology, Macao SAR

<sup>2</sup>HASLab / INESC TEC, University of Minho

<sup>3</sup>CIEd, University of Minho

<sup>1</sup> {sara.fernandes, antonio}@iist.unu.edu, <sup>2</sup> lsb@di.uminho.pt, <sup>3</sup> mhm@iep.uminho.pt

**Abstract.** It is said that due to the peculiar dynamics of FLOSS communities, effective participation in their projects is a privileged way to acquire the relevant skills and expertise in software development. Such is probably the reason for a number of higher education institutions to include in their Software Engineering curricula some form of contact with the FLOSS reality. This paper explores such a perspective through an on-going case study on university students' collaboration in FLOSS projects. The aim of this research is to 1) identify what should be learnt about software development through regular participation in a FLOSS project/community, and 2) assess the didactic potential of this kind of non-standard learning experiences. To this aim we resorted to a participatory research action approach and qualitative methods, namely case studies combining direct observation and interviews.

**Keywords:** FLOSS, Communities of Practice, Collaborative Learning

## 1 Introduction

The International Labor Organization (ILO) report [1] states that “more than 197 million people globally are out of work or 6% of the worlds' workforces were without a job in 2012”. According to Jim Lacey, CEO of Linux Professional Institute, “In 2012, 1.5 million cloud computing jobs remained unfilled due to people's lack of skills in Free / Libre Open Source Software (FLOSS) development”. In order to build the human capacity required by markets, educational systems need to prepare skillful professionals, combining a broad, informed scientific culture with sharp technical skills in specific domains. The relevance of FLOSS to the world economy singles out as one of such domains.

Due to the peculiar dynamics of FLOSS communities, effective participation in their projects is a privileged way to acquire the relevant skills and expertise in software development. Such is probably the reason why a number of higher education institutions include in their Software Engineering curricula some form of contact with

a FLOSS environment, either by exploring FLOSS projects in their laboratory activities or emulating them within the academic environment.

But there is much more to be considered. It has been observed that participating in FLOSS communities and projects provides new and unsuspected learning opportunities, with a potential added value to formal education if suitably integrated within more conventional learning contexts [2]. In this paper we explore such perspective reporting on an on-going case study on the inclusion of FLOSS projects activities in a formal learning environment. The research agenda aims at: 1) identifying what should be learnt about the practice of software development through regular participation in a FLOSS project/community, and 2) assessing the didactic potential of this kind of non-standard learning experiences. These two aims can be formulated as research questions: (RQ1) *what* and (RQ2) *how do students learn while participating in a FLOSS project and its community?* The paper is an initial report on a larger case study involving a class of pre-service Informatics teachers at University of Minho, Portugal, along the academic year of 2012-2013. Although only at a latter stage a detailed case study report and consolidated data will be available, it seems worthwhile to share the underlying pilot project and point out a number of issues already detected to call the community attention to this research. Actually, although the use of FLOSS-related projects in formal education is not new, we believe their systematic study, from an educational perspective, is still to be done.

The rest of the paper is structured as follows. Section 2 introduces the background and Section 3 the research methodology. Preliminary results are reported in Section 4; Section 5 concludes.

## 2 Background: Learning as a process

Driscoll [3] defines learning 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”. 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” [4].

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 [5][6][7], 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.

Behaviorism, cognitivism and constructivism are the three broad *learning theories* [8], most often applied in the analysis and development of instructional environments [9]. However, they pay little attention to the role of context in a learning process and, in general, the relevance of its externalities. This seems particularly critical in modern information societies, in which knowledge is permanently stored and manipulated along complex processes, which affect the way people behave and learn.

A new theory emerging in the so-called digital era is *connectivism*, which recognizes that “learning can reside outside of individuals” and “is driven by the understanding that decisions are based on rapidly altering foundations” [9]. Main assumptions are that new information is continually being acquired, learning is a process of connecting specialized nodes or information sources, and decision-making is itself a learning process.

In this framework, the paper’s starting point is the observation that communities of practice, as FLOSS development networks are, allow knowledge sharing and peer learning on a global scale and at a speed that was unforeseen one or two decades ago. Actually, the relevance of FLOSS to research in education is based on the observation that FLOSS communities may provide unusual, informal learning environments for teaching and learning software engineering skills [2]. As a viable approach to software development, FLOSS provides a model for creation of self-learning and self-organized communities, in which geographically distributed individuals contribute to build software. Well-known software applications, such as Linux, Moodle, MySQL, Firefox are good examples of the effectiveness and success of the FLOSS development model.

### **3 Research methodology**

Our research is based on a pilot project in teaching/learning software engineering. In general terms, it follows a participatory action research approach analyzed 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, behaviors 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 analyze 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 is in Social Sciences and Education. The pilot project aims to teach students, collaboratively and reflectively, 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 that is not in the 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 and unstructured interviews. 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' behavior by investigating actions and tasks that are not directly observable and fostering the externalization of motivations and expectations. All data collected is stored in the project collaborative platform hosted by Moodle. A weekly meeting of all groups with a member of the research team allows a live interaction and smoothes some difficulties in the project development.

## **4 Preliminary results and analysis**

Due to the fact that the pilot project is still running at the time of writing, only preliminary results can be reported at this point. This section focuses on two aspects 1) the project launching and 2) what students emphasize on the experience they are going through.

### **4.1 Project launch**

The 16 students (10 female, 6 male) joined the pilot project by completing a brief questionnaire on their academic background and previous experience with FLOSS. All but two older students are in their later twenties. However, the group is quite heterogeneous in what concerns academic and professional background. The latter include 3-years bachelor degrees in Computer Sciences, 5-years licentiate degrees in IT Management, or postgraduate studies in Multimedia in Education. This results in a broad range of knowledge that encompasses informatics, communication, multimedia, or management, among others.

An initial questionnaire showed that all but one of the students involved have, 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 know (or, at least, have heard about) a number of open source projects: the majority mentioned Linux, Open Office, and/ or Mozilla.

Eight groups were initially formed and invited to select a FLOSS project/community to join. Some were suggested, but each group was free to make a different choice (and they actually did). A second group choice was concerned with the role the group would play in the respective FLOSS community. Three roles were proposed and explained in a first live session: analyst, expected to document software,

programmer, to develop and integrate code, and tester. Clearly, the choice of FLOSS projects was directly influenced by the academic background. Some students, fearing of their own weak programming skills, opted for more “observational” roles, namely requirements analyst.

## 4.2 The on-going experience

For 3 months now students have been involved in the pilot project. Their global attitude has been pro-active, namely in dealing with difficulties in establishing a connection with the chosen FLOSS community. This was the main initial challenge. In few 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, was set in a mutual understanding basis, and within 30 to 40 days. This number seems too big with respect to our expectations of a live interaction with “live” communities and is a factor the research team will have to take into consideration when planning similar projects.

In all but one case, the group integration and dynamics went smoothly, probably taking advantage of a previous acquaintance between their members. Differently from what the research team was expecting, however, small groups were quickly 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 another 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 are active in using discussion groups, chats, emails, and forums to exchange ideas, doubts or achievements. They even made a number of suggestions to the research team to improve the collaborative platform. By their initiative an informal workshop, in which each group presented their own experience, was planned as a project checkpoint.

The analysis of the group reports, weekly recorded in the platform, and from live interaction in the weekly meetings, allows us to point out some initial answers to our research questions. We concentrate in just two groups for which more relevant data has been collected so far.

**Group 1.** This group joined the AnkiDroid community, a popular flashcards application for Android, and opt to act as developers ([code.google.com/p/ankidroid](http://code.google.com/p/ankidroid)). The choice was deliberate: their explicit objective was to learn programming for Android platform. The learning curve was exceptional, much steep than the one we observe in formal lectures for similar classes (RQ1).

With respect to (RQ2) students report very positively on learning-by-doing (‘we learn everyday!’). However, they were initially a bit disappointed with the gap they notice between the community expertise and their own. As pre-service teachers, they were also driven by an educational concern. Actually, flashcards are widely used as a learning drill to aid memorization by way of spaced repetition and the group wants to

promote AnkiDroid as a way to ‘mobile learning’ in their own classes. They become concerned with the practical use of the software by secondary schools students and decide to carry on a number of usability tests. This was done with specific techniques of Software Logging, to digitally record game participation, and Think Aloud, to collect verbal and non-verbal data from observation. The results of the usability tests were communicated to the FLOSS community through a live interaction in the chat and repository. The interaction with the community is understood by the group as the main driving force for their progress. Nuno, one of the students, reports in his project diary: ‘it is amazing how we are being pushed by comments from other developers. They even helped with the session [the group organized a one-day long session on AnkiDroid at a secondary school]’. During the workshop João commented on their experience: ‘Certainly not in linear way, but the guys there [the FLOSS community] are guiding our learning (...) we face so much technical problems, from the language to version control, that we are being literally forced to make progress.’

**Group 2.** Another group joined Childsplay, a collection of educational activities for young children, both as tester and developer. Differently from Group 1, the community was small and not very active. Rui reports: ‘it seems we were pushing them’ ... until the moment the group made its first contribution – a complete new game to add to the package – and became the focus of real attention from other developers.

With respect to (RQ1) the group faced a totally new programming language (python), which was mastered in less than 2 months, again with a steep learning curve. But they also acquired a valuable experience in issues like managing API versions and partial compatibility, software interfacing (with SQL) and composing third part software on the fly. This is a major point in favor of this approach: such are not the kind of skills students usually get from formal courses.

For (RQ2) the group points out the impact of being acting both as a ‘supplier’ and a ‘client’ of services in distributed development environment. Hélder comments: ‘it is the real pressure!’ and later ‘only after we delivered the first game, the “Tic-Tac-Toe”, they start taking us seriously. For the second game, the “Block Breaker”, we were even able to ask support from other developers, namely to give us nice graphics to include in the game.’

## 5 Conclusions

The preliminary results obtained seem quite positive: FLOSS-driven projects do provide a most interesting setting to exercise “learning-by-doing” and, in general, autonomous and proactive approaches to learning. It is still too early in the project to have the “big picture” which may allow extracting more general conclusions and guidelines applicable in other contexts. If our initial intuitions get confirmed, this may open a handful of perspectives for rethinking Software Engineering curricula, in particular in regions of the world where access to formal education at a university level is limited.

The results of the pilot project will be used to design an e-Learning framework to support this approach in teaching software engineering topics. The framework will then be tested in courses at various universities in following semesters.

## **Acknowledgments**

This work is supported by Macao Science and Technology Development Fund (MSTDF), File No. 019 / 2011 / A1, as well as the ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness), by National Funds through FCT - the Portuguese Foundation for Science and Technology, within project FCOMP-01-0124-FEDER-010047.

## **References**

- [1] Global employment trends 2013. *International Labour Organization*. 2013
- [2] Cerone, A. and Sowe, S.K. Using Free/Libre Open source Software Projects as E-Learning Tools. Proc Fourth Intern Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert 2010). 2010.
- [3] Driscoll, M. *Psychology of Learning Instruction*. Allyn & Bacon. 2000.
- [4] Informal Learning – An Interview with Jay Cross. Retrieved on 10 March 2013 from <http://thelearningcoach.com/elearning2-0/informal-learning-an-interview-with-jay-cross/>
- [5] Barkley, E., Cross, P., and Howell, C. *Collaborative teaching and learning techniques: A handbook for college faculty*. Jossey-Bass. 2004.
- [6] Johnson, D.W., Johnson, R.T, and Smith, K. A. *Cooperative learning: Increasing college faculty instructional productivity*. ASHE-ERIC Higher Education Reports, n°4. Washington D.C.: George Washington University. 1991.
- [7] Slavin, R.E. *Cooperative learning: Theory, research, and practice*. Allyn and Bacon. 1990.
- [8] Gredler, M. E. *Learning and Instruction: Theory into Practice*, Pearson Education. 2005
- [9] Siemens, G. A Learning Theory for the Digital Age. *International Journal of Instructional Technology and Distance Learning*. Vol. 2, No.1, 2005.