Teaching Software Engineering with Free/Libre Open Source Projects

Ioannis Stamelos, Aristotle University of Thessaloniki, Greece

ABSTRACT

One of the major problems in software engineering education is the involvement of students in real world software projects. Industry projects are a solution, but in many cases they are hard to find and student participation can be problematic due to cultural, familiarization and other practical reasons. The abundance of Free / Libre Open Source Software (FLOSS) projects is a neat solution, offering multi-lingual, multi-cultural environments in virtually every application domain, at different levels of project size, maturity, organization etc. The paper analyzes how acquisition of practical experience on several basic and advanced software engineering topics can be achieved by working in a FLOSS project. The kind of skills that can be acquired are those requested by the Overview Report for Computing Curricula by ACM and topics examined are those of the Software Engineering Body of Knowledge, by IEEE. Also software engineering areas that require special care or that may not prove suitable for such treatment are identified. Various isolated teaching cases pertaining to this approach are presented and discussed.

Keywords: FLOSS project; software engineering curriculum; software engineering education

INTRODUCTION AND RELATED WORK

FLOSS projects often provide excellent examples of well-organized, successful projects producing highly effective systems. Moreover, FLOSS projects are based on open, self-organized communities of volunteers, that manage to support software development, support and maintenance in an unprecedented way. This unique kind of virtual community provides an excellent environment for learning how to communicate with, cooperate with and ultimately learn from other members of the community. Knowledge generation and sharing (Sowe, 2006c) is implicit in the everyday operations of FLOSS communities.

FLOSS world is growing constantly. As an example, SourceForge² projects grew in number from approximately 30,000 in 2005 to 177,000 in mid 2008. However, FLOSS is not a synonym for success. Most of the new projects are one, two or three developer projects with little or no activity at all and unknown future. These, and many other bigger projects, fail
mainly because they do not manage to build a large, self sustained community.

A very interesting evolution of FLOSS in the last few years is the appearance of hybrid projects, i.e. projects that are run in a FLOSS way but at the same time sponsored by private companies. One recent study reported that one third of the 300 most active FLOSS projects nowadays are sponsored by private companies (Bonnacorsi, 2007).

From the above, it is evident that the FLOSS ecosystem provides a unique environment of numerous, open software projects and corresponding communities, building and maintaining not only software applications but also incredible amounts of knowledge. FLOSS has an evident impact on education, a fact that is already subject of analysis, investigation and discussion. FLOSS has already produced highly successful tools related to education, e.g. the course management system MOODLE\(^3\). Many reports are available on the use of FLOSS tools at various levels of education and education management, and significant experience has been already acquired (e.g. the Extremadura case reported in Bulchand, 2007). Major FLOSS distributions, such as the education version of UBUNTU (EDUBUNTU\(^4\)) specialize on education. The on-going worldwide project of One Laptop Per Child\(^5\) is currently under development and will potentially change the way education of new generations is achieved on this planet. Finally, Tigris\(^6\) is one FLOSS forge with a special focus on student projects.

However, this paper does not deal with the use of FLOSS tools while educating. Moreover, the paper does not present an approach to teach FLOSS, although by following the proposed approach, students become very knowledgeable about and familiar with various FLOSS products and processes. The paper is meant to explore the use of the FLOSS world (communities, projects) to improve higher level education, in particular Software Engineering (SE) education. Using FLOSS projects with that purpose in higher education in general is already a reality up to some extent. Therefore, we briefly review some sporadic cases of such usage of FLOSS reported in the literature. We will use knowledge from these cases to build later a full scale model. However, most probably there are many other cases that are not yet published. For example, research on the Internet\(^7\) has revealed that in one case, students at a part time Masters Degree in Computing have been asked to investigate FLOSS code and detect design patterns that were used in it. Presumably, in the close future there will be more cases of systematic FLOSS usage in software education and they will be given more visibility.

In (Jaccheri, 2007) FLOSS possibilities for empirical software engineering and software education are discussed. The paper reports on a course at the Norwegian University of Science and Technology, that is based on the involvement of students in the NetBeans project and their interaction with its community. At the Athens University of Economics and Business (Greece), in the context of a master level course titled “Advanced Topics in Software Engineering”, students are asked to participate and produce code in FLOSS projects (Spinellis, 2006). Staring et al. (Staring, 2005; Staring, 2006) also claim that “involving students in large scale, international open source projects has a potential for transformation of the relationship between students, educational institutions and society at large”. Lundell et al. (Lundell, 2007) report their experience from a practical assignment “designed to give students on an Open Source Masters course an insight into real involvement in Open Source projects” at the University of Skövde (Sweden). They also report on a reduced exercise for undergraduate students related to FLOSS. The authors found out that “the learning experience was both positive and valuable in that it gave real insight into Open Source participation”. They also report that students were further encouraged to keep on participating in Open Source projects even after their course was completed.

In (Petrenko, 2007) the issue of teaching evolution of large software systems at the Wayne State University (U.S.) is proposed to be resolved through the use of open source software and through “a software change pro-