CREATING OPEN EDUCATIONAL ENVIRONMENT BASED ON OPEN-SOURCE SOFTWARE PROJECTS

Alexander Petrenko, Vladimir Rubanov
Institute for System Programming of the Russian Academy of Sciences (ISPRAS), Russian Federation

Olga Petrenko
Moscow Institute for Open Education, Russian Federation

Keywords: Open education, Lifelong learning, Open source projects, Software engineering.

Abstract: The paper discusses principles of open education as the main method of effective education style for talented students. It is shown how open-source software development projects can naturally implement these principles for teaching software engineering courses. This is proved by examples of successful open education environments created at the System Programming sub-faculties of the two Russian top-ranked universities - Moscow State University and Moscow Institute of Physics and Technology (Phystech). These sub-faculties are run jointly with the Institute for System Programming of the Russian Academy of Sciences, the leading Russian research & development government organization in the field of software engineering.

1 INTRODUCTION

Educational system should match the needs and tendencies of continuously developing society to sustain its evolution. Traditional system of professional education was mainly based on giving to students some specific knowledge and skills that can be applied in some practical work at once. However, under the conditions of intensive technological evolution, such an approach to education becomes irrelevant because concrete practical knowledge quickly becomes outdated and unclaimed while the main abilities demanded in these conditions become adaptability, constant knowledge update, decision making unbiased from established patterns, dynamic activity planning, etc.

Schematically, the main fields of skills and knowledge of IT-professionals can be presented as in Fig.1. So, just subject knowledge is not enough for successful work of a good specialist in the modern society. The Memorandum of European Commission on lifelong learning emphasizes the need in such social skills as acting with confidence, result oriented focus of own actions, right balancing of risks and responsibility in decision making as well as such cognitive skills as ability to constantly learn, adaptability to changing environment, skills to dig for right information in various areas and ability to filter necessary information in the huge informational flow that each active individual in the modern society is subject to.

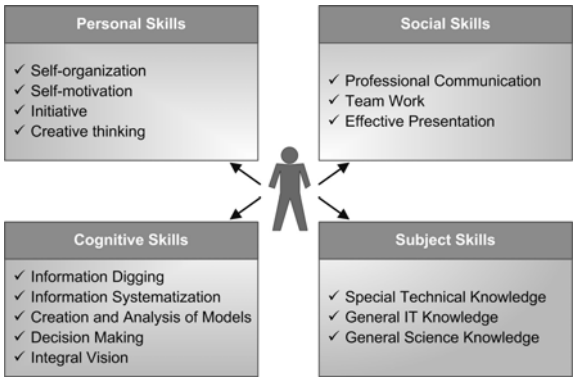

Figure 1: Main skills of an IT-professional.

Acknowledging these principles of continuous lifelong learning stimulates transition of educational systems towards so called *open educational* system. This system is mainly oriented on upbringing independent self-motivated individuals that are able to effectively collaborate with the quickly changing

Petrenko A., Rubanov V. and Petrenko O. (2009).
CREATING OPEN EDUCATIONAL ENVIRONMENT BASED ON OPEN-SOURCE SOFTWARE PROJECTS.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 235-238
DOI: 10.5220/0002159002350238

world - individuals that can and want to effectively learn constantly rather than just apply known and established practical skills in some steady work.

In such open educational systems, educational opportunities are open to students - they can see paths of possible education and evolution and thus it becomes possible to discuss with its mentor the means and specific actions for achieving individual goals in the broad space of these opportunities. One of the main characteristics of open education is responsible decision making by the student about the needed educational goals and means to meet them. Students have to realize and take full responsibility in this.

The main principles of open education are:

- individualization of education;
- responsibility for own success;
- collaboration;
- continuous learning.

To enable transition to the open education style, it is worth doing the following:

- It is necessary to teach students to be self-dependent in education. Being active is a key thing for this.
- Students must be involved in the mutual personal communication.
- The starting and the target points of each particular part of the educational path should be individual.
- Explicit stage of reflection should be introduced in the education process.

The important general step towards open education is creation of special self-managed educational environment that stimulates students to be actively involved in leading professional societies in the studied field and to actively communicate and collaborate within them. Such societies provide various opportunities in implementing educational goals and students can freely and independently (often unconsciously) choose whatever fits them best individually. The importance of such environment was mentioned in the historical letter of Nobel Prize winner academician P.L. Kapitsa to Iosif Stalin in 1946 about establishing the legendary Phystech (Moscow Institute of Physics and Technology). The idea was to gather most talented students from all around USSR and invite the best academic and industry specialists to teach them in a new revolutionary at that time educational environment, which was then called “Phystech System”. Kapitza wrote: “From the very first steps, educating talented students in the atmosphere of technical research and freedom of creativity using the leading laboratories of the country”.

In this paper, we present the modern rethinking of these principles and show how they can be implemented for upbringing top-level specialists in the field of software engineering by using open-source projects combined with proper courses and mentors to build open educational environments. The first chapter discusses the benefits that open-source projects can provide for building such environments. The second chapter presents practical experience of using open source projects at the System Programming sub-faculties of the two Russian top-ranked universities - Moscow State University and MIPT (Phystech). These sub-faculties are run jointly with the Institute for System Programming of the Russian Academy of Sciences.

2 OPEN-SOURCE PROJECTS FOR OPEN EDUCATION

Using open-source software development projects can be a good basis for building open educational environment in the field of software engineering. Such projects can effectively and easily enable the following tools and educational means for almost any specific technical subject:

- *Ready educational materials*: requirements for building software, the executable software itself, specifications of its architecture, implementation source code, lists of already fixed and still standing problems and features elaborated at the development and maturity phases of the product lifecycle.
- *Ready infrastructure for practical classes*: configured informational systems for version control and project collaboration with possibility to communicate with real developers of the product.
- Possibility to create *strong motivation* for students by involving them in the practical activities in real life developing projects, especially if such projects have high social importance and prestige.

The diversity of open-source software projects (which is drastically increasing in the recent years) delivers to students a wide variety of choice options for thinking and deciding individual learning paths within the general software engineering educational plan. The materials of open-source projects can serve as a public easily accessible through Internet

knowledge base in a specific subject - in some sense, a counterpart of the classic public libraries that help students to increase independence and activity in the educational process as well as to individualize it. Also, open source projects provide wide variety of choices of technical aspects such as programming languages, development technologies and methodologies even within a specific topic because, usually, there are a number of different competing projects that use different technical approaches for implementing the same functionality or a product of the same kind. This variety and freedom of choice improve motivation and responsibility of students quite in line with the principles of open education.

3 PRACTICAL OPEN EDUCATION IN SOFTWARE ENGINEERING

In the recent years, open-source software development projects became an important part of the educational process at the System Programming sub-faculties of the two Russian top-ranked universities - Moscow State University and Moscow Institute of Physics and Technology (Phystech). Under the established collaboration schema, all specialized classes and practical sessions at these sub-faculties are driven by employees of the Institute for System Programming of the Russian Academy of Sciences (ISPRAS), the leading Russian research & development government organization in the field of software engineering. High-year students come to ISPRAS premises to perform their research and development practice and learn special courses while taking all the other general classes at their "alma-mateur".

A few years ago we [ISPRAS] decided to introduce a requirement for students to participate in at least one public open source project that they can choose at their own. Also, we changed classical system of theoretical software engineering classes and practical sessions to a more flexible system under the principles of open education. Each student is assigned a personal mentor from employees of ISPRAS who keeps in close touch with the student, answers his/her personal questions and advises useful informational sources. One mentor may serve no more than 4 students. Participation in the real life open-source projects supplements theoretical courses and classic practical sessions that are based on pre-selected model examples and tasks. The main new stages of the new system include:

- *Choice of an open-source development project*. The project must have a public infrastructure (usually at http://sourceforge.net) and be active (that is there should be on-going developments). Students may also join any open-source projects that are performed at ISPRAS itself. Students evaluate various projects by themselves, mentors just check if the final choice meets all the necessary requirements.
- *Acquaintance with the project*. Students study their selected projects on their own with the focus on those aspects important for their personal educational path and for the specific courses at the sub-faculty. At the end of this phase, students make *presentations* (more correctly *seminars* or *defenses*) of their projects to the classmates and mentors where they can be asked any questions by the audience. Mentors ensure that the students can make correspondence between the theoretical information taken from courses with specific practical aspects of their projects by proper questions. Additionally, students are asked to prepare written reports with a defined structure.
- *Practical tasks*. Mentors prepare a number of tasks for students with some points assigned to each task. Students may choose a number of tasks to do to reach a defined target in terms of these points. Students perform regular presentations of their work similarly to the presentations at the acquaintance phase.

Usually, students do particular improvements in their projects such as analysis of existing and elaboration of new specific requirements for the project's product, performing architectural design of their possible implementation and finally implementing them in code, testing and writing corresponding documentation. The criterion for success of such activities is inclusion of the changes and improvements into the real project mainline. Mentors encourage students to directly collaborate with the real project team and use feedback of the team as indicators of successful direction of their work. This strongly improves students' motivation as the feedback from external parties is perceived by students as more objective than by the internal people.

However, participation in the real open-source projects alone is not enough for implementing all the principles of open education. It is the continuous comparison of the practice in the project work with the theory being studied in parallel that drives the effective educational process. So it is this symbiosis

that helps implementing the effective education in the open style. The presentations and communication with classmates and mentors also play an important role to complete the open educational environment. Students are asked (and thus encouraged to notice and learn) not only technical questions but also questions on organizational aspects such as project team structure, roles, deadlines, quality issues, business perspectives, etc.

Based on our practice, at the first presentations, students are usually in almost complete confusion because at the first years at their universities they get used to the traditional system of fixed information to learn with subsequent exams so that they are always directed what to do. Only after a while, after working in the more flexible and open environment, they start realizing the real diversity of the problems in projects and the importance of proactive position of participants to solve them. At the end of the day, this helps to understand and learn all the principles of software engineering at the substantially higher level.

4 CONCLUSIONS

In the modern society and the fast changing world, principles of open education are acknowledged as key factors for upbringing elite professionals. This applies to IT-professionals in particular. Implementing these principles for educating students can be done in a number of ways. At our organization [ISPRAS], we promote the use of public open-source projects as the basis for learning software engineering in the open education style. Such projects provide good infrastructure for ready to use educational materials, technical and organizational structures and development communities capable of providing real life feedback. Leveraged by individual mentors from the industry and individual learning paths both in for students' practical work and specific theoretical courses, this proves to be a very efficient approach for teaching software engineering in a way that can turn even conservatively taught students into skilled and, most important, self-motivated and mature active professionals.

There are a number of points worth to mention though. Initially, the only requirement for self choosing open-source projects by students was the publicity and active development stage of the projects. The additional requirements that we eventually came to later became the size of the project and level of maturity because small (less than 30000 lines of code) projects or projects run by home-made enthusiasts proved to be irrelevant for demonstrating all the complex aspects of industrial software engineering. The projects do not need to be ideal in organizational and technical excellence (analyzing and dealing with drawbacks helps educating students as well) but they should still be good enough to be useful. Regular meetings like presentations and seminars where group of students and their mentors communicate and discuss various issues from both the projects and theoretical courses in a free way is very important factor to mitigate educational risks and to prevent mislearning.

The presented approach has being successfully used at the System Programming sub-faculties of the two Russian top universities - Moscow State University and MIPT (Phystech), where specialized education of high-year students (including their diploma work) is performed at our R&D organization of the Russian Academy of Sciences.

REFERENCES

Commission of the European Communities, 2000. *A Memorandum on Lifelong Learning.* Commission staff working paper.

U.S. Department Of Labour, 1991. *A Scans Report for America 2000. What Work Requires of School.* The Secretary's Commission On Achieving Necessary Skills.

W. Huitt, 1997. *The SCANS report revisited*. In Proc. of the Fifth Annual Gulf South Business and Vocational Education Conference, Valdosta State University, Valdosta, GA.

V. Kuliamin, V. Omelchenko, and O. Petrenko, 2005. *Active Learning Facilitates Success of Formal Methods in Practice*. In Proceedings of SEEFM'2005, Orchid, Macedonia.

V. Kuliamin, V. Omelchenko, and O. Petrenko, 2006. *Forming professional competencies of the modern software developer*. In Proc. of ISPRAS, vol.9, pp. 237-250.

A.Petrenko, O. Petrenko and V.Rubanov, 2008. *Combining Open Education and Open Source Projects*. In Proceedings of the 3rd Conference on Open Source in Education. Pereslavl-Zalesskii, Russia.