On the Importance of Dialogue with Industry about Software Engineering Education

Letizia Jaccheri
Department of Computer and Information Science
Norwegian University of Science and Technology
7491 Trondheim, Norway
letizia@idi.ntnu.no

Sandro Morasca
Dipartimento di Scienze della Cultura, Politiche e dell’Informazione
Università degli Studi dell’Insubria
Via Valleggio 11, I-22100 Como, Italy
sandro.morasca@uninsubria.it

ABSTRACT
This paper is based on the experience gained in several iterations of software engineering courses held by the authors both at universities and in industry. The paper provides some reflections and lessons learned that can be used for designing or reflecting about software engineering education. The main contribution is the definition of five possible roles that industry can play in software engineering education from the point of view of the university teacher.

Categories and Subject Descriptors
D.2 [Software Engineering]: Management; K.3.2 [Computer and Information Science Education]: Computer science education; K.6 [Management of computing and information systems]: Project and People Management—Training

General Terms
Experimentation, Human Factors

Keywords
Software Industry, Software Professionals

1. INTRODUCTION
In this paper, we address the problem of making software engineering education relevant to industry [6] [8].

Paraphrasing [3], the questions that teachers should always have in mind when interacting with industry are: Do our software engineering research and the education content we provide to our students produce the knowledge that today’s professionals can apply in their daily work? Do our research and our teaching address the problems or challenges that are of concern to the software industry? Do we focus on current technological and business issues? Is our teaching material accessible to the software industry?

The topic of educating software engineers from an industry point of view is addressed in [6]. [6] concludes that the main skills that industry requires from new hired persons are: team work, testing and evaluating capabilities, effective communication skills, quality measurement, and process improvement.

Configuration management, testing and quality assurance, process standards, maintenance and reengineering, project management, object oriented analysis and design, user interfaces and human-computer interaction, and requirements gathering are among the topics which are of industrial importance by the survey reported in [8].

In this work, we structure the experience we have accumulated by interacting with industry as software engineering teachers and researchers by providing a classification of the roles industry can play with respect to software engineering education. Specifically, we have identified (see Section 2) five main roles that industry can play in software engineering education. This list of roles is not exhaustive, but can be used as an initial set that can be refined and extended in future work. Section 3 concludes the paper.

2. ROLES OF INDUSTRY IN SOFTWARE ENGINEERING EDUCATION
We observe that industry can play at least five roles in software engineering education:

- Industry as students. This is the case of industrial training or continuous education, and it may be the case of the participation of industrial professionals in university software engineering classes.
- Industry as former students. According to [8], this is a way of being in dialogue with industry to assess their knowledge needs.
- Industry as teachers. This happens when professionals are invited by academic institutions to share their expertise with students.
- Industry as researchers. This happens when empirical investigations are run in the context of software engineering courses and industry are interested in the research results.
- Industry as customers. Industry can play the role of the customers for the projects in software engineering classes.
2.1 Industry as students

The software industry perceives training as one of the most important activities for its personnel. Due to the continuous evolution of software engineering, software companies and their employees are almost forced to keep abreast of the competition by continuously updating their knowledge on the field especially on specific topics. A number of larger-size software companies have organizational units specifically devoted to organizing software engineering courses. Other companies, usually smaller-size ones, send their employees to classes organized by larger companies or by educational companies whose mission is to organize and market software engineering or ICT-related classes.

Industrial professionals usually have a different mindset and goals than college students. Industrial professionals are more interested in learning specific methods and techniques that can help them in their day-to-day activities. As an example, professionals may be more interested in learning how to use UML and a UML specific tool than in learning about the general principles of Object-Oriented software engineering. This approach has the obvious advantage that software professionals may obtain a lot from a well-focused course. On the other hand, the disadvantage is that a number of software engineering methods are more fundamental ones and are not necessarily directly related to some specific technique. Their main benefits will be had in the long-term. So, a balance must be made between courses that provide short-term and long-term benefits.

Teaching courses to industrial professionals allows software engineering teachers to have a better sense of what is important to the industry and also how it can be taught in college classes. A number of differences exist between the relationships of the instructor and the industry professionals that attend classes, and the relationship between the instructors and college students. Industry professionals have a number of demands that usually may be very specific, while college students have more general expectations. As a matter of fact, industry professionals’ backgrounds, demands, and expectations may be very diverse. This may pose a number of problems to the instructor, but it will also give him or her the chance to get in touch with a wide spectrum of needs that software professionals may have. Also, industrial professionals have a different approach with the instructor. Unlike college students, industrial professionals will usually not be judged by the instructor. Thus, it is usually possible to obtain good feedback from the participants in these courses on what should be taught and how. This feedback needs to be processed by the teachers and used in the continuous evolution of software engineering courses.

2.2 Industry as former students

[8] reports about a survey that has been run among computing practitioners with the goal of assessing the relevance of their education as former students. In this case, the dialogue with former students is implemented by use of surveys and formal analysis.

Former students, especially those who have been a teacher’s degree candidates are an excellent communication channel between a teacher and industry. In several cases, software professionals have carried out their first industrial-strength project during their dissertation work, and that may have helped them begin a career in an industrial organization.

After graduation, there certainly is a change in the perspective of both former students and teachers. The former student may discover that the teacher’s professional interests go well beyond the specific subjects of the dissertation or those taught by the teacher during college courses. This allows the communication to be broader and change its scope over time, with the evolution of the technologies and the industrial needs. On the other hand, teachers are less constrained in the scope and time horizon of their research and teaching activities than practitioners are in their industrial activities. So, former students may look to their advisors and teachers for longer-term perspectives and alternative solutions to the ones they already utilize. Teachers also carry out research and in some cases are also consultants, so they may bring ideas that a software company may be willing to take into account and explore. All of this helps shows the industry that the research and the educational activities carried out in universities may have practical effects and give the software professionals a more complete picture on those and why some topics are taught in software engineering education.

From the viewpoint of the teacher/researcher, a close contact with a former student may make the interaction smoother and easier, so the teacher/researcher will have a somewhat “privileged” channel with the software industry that can be used to identify “hot” topics and trends that can be used in his or her research and teaching activities.

2.3 Industry as teachers

Industrial professionals may provide a great value added to conventional college classes. The experience of software professionals usually provides a complementary view to the issues and problems of the real world to that often offered in college classes. Professionals certainly bring a different perspective than the teachers’ and allow the students to realize the importance of a number of issues that they may not fully appreciate. As a simple example, it is typical of software engineering students to believe that the coding phase is the most important one. After all, that is the phase in which the “real” product is created and it is also the phase that is used in the first software classes, which are usually about software programming. Thus, it may be difficult for the teacher to convince the students that other phases (e.g., the early ones) may be more important, and that, for instance, it is more important to invest more time in devising a good software architecture than start programming right away. More generally, it may be difficult for the teacher to show the industrial aspects of software production. The students are usually more interested in the technical ones, and often view managerial and organizational aspects as marginal. Software professionals may be far more convincing for the students because they bring real-life experiences with all the relevant pieces of information, technical and organizational. They can explain the rationale behind decisions, which may have been based on several factors in addition to and beyond merely technical ones. The students are usually very receptive and eager to hear what a representative of the “real world” teaches.

Software professionals may participate in teaching activities in several ways and with different degrees of involvement, from one-time talks in the context of existing classes to teaching entire courses. Especially when they teach entire classes, it is important for software professionals to have an idea of the relationships of the class they are teaching with...
2.4 Industry as researchers

Unlike many other engineering branches, software engineering has often proceeded in a mostly ideological way. New methods and techniques have hardly ever been accepted into industrial practice based on evidence. There are a number of reasons for this, but this just could not be possible in other engineering branches, where the gathering of evidence is an integral part of the scientific method and of the technological process.

Nevertheless, software engineering has been built and has evolved through theoretical and applied research. It may be believed that universities carry out the majority of theoretical research while applied research is mostly carried out in industrial environments. Even though this may sound like a blanket statement, there is some truth to it. Few universities in the world have the necessary resources to carry out applied research that leads to the in-house development of successful, market-oriented software products. On the other hand, industrial organizations, with their primary interest on being profitable and the time pressure under which they work, certainly have no time and resources to devote to theoretical investigations, which may take too long to yield practical results, if any.

However, industrial organizations would certainly have a competitive edge if they could obtain new methods and techniques from theoretical research along with evidence about how effective they are or how they should be used. The same actually applies to existing methods and techniques, which are usually introduced with a few guidelines that are rarely based on any kind of evidence. It is our position that the transition phase between theory and its application in industrial settings should include evidence gathering too. In other disciplines, preliminary evidence is collected via laboratory empirical studies.

We could do the same in software engineering, by running “lab” experiments. As software engineering is a human-intensive business, laboratory studies may involve people. One possibility is to carry out lab experiments with students in the context of software engineering courses. As we explain in [5], there are a number of benefits and costs for students, teachers, researchers, and industrial organizations. We also highlight that software companies need to be a driving force behind these empirical studies. They will really benefit from the results only if they get extensively involved in the experiments, from the initial idea about the focus of the specific experiments down to the interpretation of results and their practical use. In return, software companies will obtain an evaluation of methods and techniques that may be used in several ways. At a minimum, the lab experiment can be used as a pilot study on which further experiments can be based in the industrial organization. At the other end of the spectrum, the results of the lab experiment can be used as the evaluation of the methods or techniques.

On the other hand, students will have further ideas about industrial practices, since it would be in the interest of the software companies to run experiments in environments that resemble their work environments as closely as possible. In addition, the alternative for them would be to attend traditional lectures and exercises or carry out somewhat artificial projects on the subject of the experiment.

The role of the teacher is to make sure that the students receive enough educational content from the experiments and the experiments fit right in the framework of the course. This, the teacher needs to find a satisfactory trade-off between the research and industrial needs, on the one hand, and the pedagogical needs on the other hand.

As for the researchers’ and industry’s sides, the so-called “external validity” of the results is an issue, since laboratory studies will involve students and not software professionals. So, the problem is: how applicable are the results for software companies? There is obviously no final answer to this, but at the very least having some evidence on a technique is more useful than having no evidence at all. Then, it is up to the software organization to make sure that the experiment is run in such a way that its results are interpretable and applicable in its own context. It is also worth noting that the distinction between students and software professional is becoming more and more blurred in several cases. Students have part-time or full time jobs or work as consultants. Professionals go back to college to improve their education.

2.5 Industry as customers

The software engineering education literature contains several instances of articles about team project courses and disciplines required to work on software teams. The community [2] describes a software engineering body of knowledge and curriculum model and advocates for the importance of including software projects as part of software engineering core area. For example, [9] describes a framework for teaching software project courses. [4] describes an approach to teaching in a simulated industrial environment. We believe that it is very useful for students to be taught the concepts and techniques of a discipline like software engineering not only by traditional lectures and somewhat artificial, academic exercises, but with hands-on practice that resembles industrial practice as much as possible. Thus, it is desirable for the assignments given in software engineering classes to have industrial relevance, even though it is hardly ever possible to give a full-fledged industrial project as a course assignment to students. However, industrial organizations may provide significant subsets of requirements of their application, so they can be used in several ways during college classes, as

- homework assignments: this is the case when the subset of requirements is too large to be used during class hours or when students are required to carry out some additional applied work beyond class hours;
- projects to be carried out during class hours: ideally, one industrial-strength project may be used to provide the motivation and the examples for the set of concepts and techniques explained during a software engineering class; besides the technical aspects, the teacher can also talk about the relative importance of the different techniques and how they can be combined; also, it is possible to motivate and compare different solutions for the same problem and show how the problem was addressed and solved in a real industrial environment;
• examples: even if it is not possible to use one industrial-strength as an ongoing case study throughout a software engineering class, examples may be taken from it to explain various concepts and techniques.

In [7], we report about one instance of this kind of interaction with industry. Another example at the Norwegian University of Science and Technology is reported in [1]. An industrial organization therefore plays the role of the customer, and, as such, some degree of cooperation is required, the extent of which may vary from case to case. As we described in Section 2.3, industrial professionals may play the role of the teachers as well, and that would help provide guidance to the students.

3. CONCLUSIONS

When designing and implementing courses for practitioners, teachers are encouraged to focus software engineering content on the knowledge that today’s professionals can apply in their daily work. When inviting professionals to teach to our students, teachers are forced to see the content of our university courses in light of the problems and challenges that are of concern to the software industry. In this process, teachers must make an effort to place theoretical knowledge in the context of current technological and business issues. To be able to interact with industry about our courses and projects in which industry may play the role of the customer, we have the challenge of making our teaching material accessible to the software industry.

In this short paper we look at five different roles that industry can play with respect to software engineering education seen by our point of view that is that of software engineering teachers and researchers. To sum up, the dialog with industry is important to increase industrial relevance [3] of our teaching and research.

4. ACKNOWLEDGMENTS

The research presented in this paper has been partially funded by the IST project “QualiPSo,” sponsored by the EU in the 6th FP, the FIRB project “ARTDECO,” sponsored by the Italian Ministry of Education and University, and the project “La qualità nello sviluppo software,” sponsored by the Università degli Studi dell’Insubria.

5. REFERENCES