Chapter 45
From Teaching Software Engineering Locally and Globally to Devising an Internationalized Computer Science Curriculum

Liguo Yu
Indiana University South Bend, USA

ABSTRACT

Software development is the process to produce an information technology solution to a real-world problem. Teaching and integrating non-technical software engineering skills into the curriculum is considered one of the most challenging tasks in an academic environment. This becomes even more challenging when the curriculum is supposed to be internationalized and applied in different countries because of the cultural difference, policy difference, and business model difference. In this chapter, the authors present their experience of teaching a software engineering course both locally and globally, where two universities of USA and China are chosen for this study. Specifically, they describe how they adjust homework assignments and student performance evaluations to reflect different government policies, different business environment, and different real-world customer requirement. The chapter shows that it is possible to create an internationalized computer science curriculum that contains both common core learning standards and adjustable custom learning standards.

INTRODUCTION

Curriculum internalization is a new trend in education. More and more internationalized curricula are being created. The benefits of globalizing education are enormous. However, risks and challenges also exist. This is especially true for engineering education, such as software engineering education, where knowledge and skills should be adapted to local policies and local business environment.

DOI: 10.4018/978-1-7998-3016-0.ch045

Software engineering is a discipline in computer science. Software engineering education is the process we educate computer science students or information technology students with required knowledge and skills to compete in the software industry. The learning objectives of software engineering not only include technical skills, such as modelling, design, and programming, but also non-technical skills, such as communication skills, management skills, documentation skills, and customer interaction skills (Sedelmaier & Landes, 2014; Kuhrmann et al., 2014; Johns-Boast, 2014). Generally speaking, these non-technical skills are hard to deliver in a traditional lecture-based classroom, where it is not effective in providing hands-on experience for students. Instead, problem-based learning (Hmelo-Silver, 2004; Hung, 2011; Kay et al., 2000) has been used in software engineering education to achieve these learning objectives.

Problem-based learning allows students to learn software engineering knowledge and skills through solving a real-world problem. This is in contrast to the traditional lecture-based learning. Under the problem-based learning environment, although instructors no longer need to spend their majority efforts on lectures nor directly participate in the problem-solving process, they are still the key players. Their main tasks include designing the problems, assisting/coaching students, and evaluating students' performance. The new tasks have raised new challenges for instructors, some of which are listed below:

- How to design an assignment so that it is more similar to a real-world problem?
- How to help students solve a problem so that appropriate assistance should be given to students without suppressing their creativity?
- How to evaluate students' performance if the assigned problem is not solved or not completely solved?

Implementing problem-based learning in an internationalized curriculum could be even challenging. First, real-world problems have their own context; they are closely related to business models, government policies, and cultural environment; some problems considered critical in one country might be irrelevant in another country. Second, we are living in a dynamically changing world; problem-based learning in an internationalized curriculum should be revised frequently to reflect the transforming global reality (Kahn & Agnew, 2015).

In this chapter, I present my experience of applying problem-based learning to teach software engineering in two universities of two different countries, Indiana University South Bend, USA and New York Institute of Technology-Nanjing, China. The business environment differences between the US and China provide valuable insights on how to adjust teaching strategies locally in order to design a potential internationalized curriculum.

The remaining sections of the chapter are organized as follows. I first present the ideas to create an internationalized computer science curriculum. Next, I review the background knowledge. Then, I describe the software engineering classes which are taught both locally and globally. Finally, conclusions are presented.

COMPUTER SCIENCE CURRICULUM INTERNATIONALIZATION

Computer science and software engineering education, like other engineering educations, could be standardized and internationalized. This is mainly because the software development process is already standardized with CMM model and ISO 9000 model. Software organizations around the world are