Chapter XV
Continuous Curriculum Restructuring in a Graduate Software Engineering Program

Daniela Rosca
Monmouth University, USA

William Tepfenhart
Monmouth University, USA

Jiacun Wang
Monmouth University, USA

Allen Milewski
Monmouth University, USA

ABSTRACT

The development, maintenance and delivery of a software engineering curriculum present special challenges not found in other engineering disciplines. The continuous advances of the field of software engineering impose a high frequency of changes reflected in the curriculum and course content. This chapter describes the challenges of delivering a program meeting the needs of industry and students. It presents the lessons learned during 21 years of offering such a program, and dealing with issues pertaining to continuous curriculum and course content restructuring, the influence of the student body on the curriculum and course content. The chapter concludes with our recommendations for those who are seeking to create a graduate program in software engineering, with a special note on the situations where an undergraduate and graduate program will need to coexist in the same department.

INTRODUCTION

The objective of this chapter is to prepare those who are seeking to introduce a graduate program in software engineering (SE) for the challenges they will face. Towards that end, the lessons learned during 21 years of offering such a program at Monmouth University will be presented. As it

will be demonstrated, the development, maintenance and delivery of a software engineering curriculum present special challenges not found in other engineering disciplines.

This chapter describes the challenges of delivering a program that meets the needs of industry and students in a highly dynamic field. The evolution of the curriculum induced by the domain's continuous advances and evolution in industry practice will be presented. The special meaning of continuous course content development in software engineering will be argued through issues pertaining to dated textbooks, ever-changing programming languages, operating systems, and software tools. The chapter will also present our experience in dealing with the diversity of the student body, and its influence on the curriculum and course content. The chapter will conclude with our recommendations for constructing a similar program, with a special emphasis on situations where an undergraduate and graduate program in software engineering will need to coexist in the same department.

BACKGROUND

Although software engineering was recognized as a field in 1968 at the NATO sponsored conference on the subject (Naur, 1968), it took universities and colleges a significant amount of time to respond to that fact. It was not until 1986 that Monmouth University (MU) started a graduate program dedicated to software engineering, which was offered by its Computer Science Department. In 1995 Monmouth created the first Software Engineering Department in United States. Now it is one of the pioneer universities offering a bachelor's degree in software engineering.

One motivation for creating a separate software engineering program and department was the awareness of the skills that industry would like students to have upon graduation, which are not stressed by most computer science curricula. These skills include teamwork, communications, time management, engineering problem solving, quantitative and qualitative process management, reuse, requirements management, system architecture, testing and project management.

As one of the few universities with extensive and comprehensive experience in offering software engineering programs, we have learned much about providing such a program. With more and more undergraduate software engineering programs appearing, we feel it is beneficial to other institutions for us to share the problems encountered and lessons learned over the past 21 years. A summary of the problems encountered and the lessons learned are presented here:

- **Continuous curriculum restructuring.** One can expect to revisit the overall curriculum of the program every four to five years, in order to accommodate changes in industry practice and educational expectations. This is reflected also in the historical investigation of the graduate software engineering curriculum reported in (Duggins, 2002).
- **Continuous course content restructuring.** It is critically needed due to the dynamics of the field. The continuous development of course content implies also a continuous development of course projects, and dealing with dated textbooks, ever changing operating systems, programming languages and software tools.
- **Hiring and retaining faculty.** The need for new faculty to have a record of sustained scholarly accomplishments and industrial experience enforces great restrictions on the number of available candidates, as it was also notified by Glass (2003). Retaining faculty is complicated by the fact that in addition to performing their normal teaching duties SE faculty must continually keep up with changes in the field as a whole.
- **Influence of the diversity of the student body on the curriculum and course**