Computer Science Education
2001, Vol. 11, No. 1, pp. 3–5

0899-3408/01/1101-0003$16.00
© Swets & Zeitlinger

Practical Software Engineering Education

Hossein Saiedian
Department of EE & Computer Science, University of Kansas, Lawrence, Kansas, USA

Those of us who have been in the field of computing, and in particular computer science and software engineering, have been privileged to have a relatively stable body of students interested in registering for our classes and programs. While there has been much motivation and interest on the part of students to consider our classes and programs, we have had and continue to have a major responsibility: maintaining and intensifying the initial interest. One effective solution has been to present our computing and technological ideas in their greater social and practical context. In this case, teach and train our students to be most effective in solving real software engineering problems.

Furthermore, some studies have shown that the state and quality of the software engineering workplace is a direct function of the quality of software engineering education (Beckman et al., 1997). Thus we have a more serious responsibility: not only maintaining our students' interest, but through them make an impact on the state of software engineering practice. That is the theme of this special issue of *Computer Science Education.*

The articles of this special issue (except for the last one) were selected from the papers presented and published at the 13th Conference on Software Engineering Education and Training (CSEE&T) which was held in Austin, Texas (USA) during March 6–8, 2000. The conference, as its title clearly reflects, is about software engineering education and training and continues a tradition of providing an excellent forum to discuss related topics to promote innovation and collaboration, and to stimulate new instructional approaches to software engineering education and training. All selected articles, which have been modified and extended from their initial versions, address the same issue: how to present software engineering concepts in their most effective practical context? The authors of the selected articles have done an excellent job in addressing this issue. The last article, which is an invited contribution,

Correspondence: Tel: +1(913)897-8515. E-mail: saiedian@eecs.ukans.edu

addresses a more general and very related topic: software engineering as a professional discipline. A short summary of each article is given below.

Can two programmers, working collaboratively together on various aspects of a software project, be more productive than working alone? Laurie Williams (North Carolina State University, USA) and Robert Kessler (University of Utah, USA) look at this issue and share their research and observations which are based on two courses taught at the University of Utah. The approach, called pair-programming, is discussed and the influences of positive pair pressure and pair learning are presented. In addition to the benefits gained by the pair programmers, Williams and Kessler observe that the workload of teaching faculty is also reduced because the pair programmers often seek support and advice from one another.

Ray Vaughn of Mississippi State University (USA) presents his experiences in teaching industrial practices in an undergraduate software engineering course. The students are placed in an industrial working environment to develop a software project for a real client. He discuses successes, failure, and opportunities for improvement. In particular, he discusses team dynamics, planning, accountability, on-time and within-budget delivery, and product roll-out. Supporting materials are also provided in the article.

Groupware support for the software engineering students is the focus of the third paper by Sarah Drummond, Cornelia Boldyreff and Magnus Ramage, all with the University of Durham (UK). They argue that software engineering teams will work more effectively if supported by network-based groupware technology. Their research investigates the provision of such groupware support and their potential impacts. The authors' experience with a particular tool to support group work is explained. At the end, the authors provide an initial quantitative assessment of the student feedback.

Is the management of software project more difficult than management of other projects? The fourth article by James McDonald of Monmouth University (USA) studies this issue and concludes that software project are in fact more difficult to manage. Several factors are discussed and the author explains how these factors will negatively impact the software projects, especially in areas such as estimation, scheduling, and planning. Guidelines for addressing each of these difficulties are provided in the article and topics that need to be incorporated into the software project management courses to minimize the impact of such factors are discussed.

The last article is a collaboration between Don Bagert (Texas Tech University, USA) and Nancy Mead (Software Engineering Institute, USA) to investigate software engineering as a professional discipline. The authors discuss what may constitute software engineering as a legitimate engineering profession. Topics such as licensing, certification, and accreditation are discussed in detail. The authors also present an excellent look at historic trends, current status, and future outlook of each of these areas and conclude that this will be an area of professional interest as the software engineering field will mature.

Putting this special issue of *Computer Science Education* together has been an enjoyable process. I'd like to thank the authors for submitting their work and the reviewers for their excellent evaluations. Working with the *Computer Science Education* editors, Renee McCauley and Sally Fincher, has been a pleasure. Renee McCauley has been especially patient, supportive, and helpful. I also would like to thank the Organizing Committee of the 2000 Conference on Software Engineering Education and Training, and especially Susan Mengel, for putting together an excellent program for the conference. Enjoy the special issue!

REFERENCES

Beckman, K., Coulter, N., Khajenouri, S., & Mead, N. (1997). Collaborations: Closing the industry—academia gap. *IEEE Software*, *14*(6), 49–57, 11.