ELSEVIER

The Journal of Systems and Software 49 (1999) 113–115

The Journal of Systems and Software

www.elsevier.com/locate/jss

Software engineering education and training for the next millennium

Hossein Saiedian [*],[1]

Department of Computer Science, University of Nebraska at Omaha, Omaha,Nebraska 68182-0500, USA

Received 6 April 1999; accepted 6 April 1999

The quality of the software engineering workplace is a direct function of the quality of software engineering education (Beckman et al., 1997) although other factors may play a role. Thus proper software engineering education can significantly improve the current state of software development and help alleviate many of the traditional problems and crisis associated with industrial software practices (Gibbs, 1994). An education and training that includes not only the basic knowledge and skills of computer science but focuses on the definition, development, and maintenance of software, will help our efforts in improving the quality of software produced by the next generation of software engineers.

The Conference on Software Engineering Education and Training (CSEE&T), 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. The 1999 conference (CSEE&T'99) brought together some of the most active researchers and practitioners to describe the best practices and innovative approaches in software engineering education and training in three intensive days of paper presentations, panel discussions, and workshop sessions. Each day of the conference focused on a different theme. The themes were as follows:

1. *Professional Issues* (e.g., Accreditation, Licensing, Ethics)
2. *Training Curricula and Distance Education*
3. *Undergraduate and Graduate Curricula*

We had the privilege of presenting three excellent keynote speakers: Leonard Trip (President, IEEE-CS, 1999), Don Gotterbarn (Software Engineering Ethics Research Institute), and Doris Carver (Louisiana State University) who represented the industry as well as the academia. The papers, panels, and workshops forming the the technical program explore software engineering education and training in a wide variety of contexts. Panel discussions were an integral part of the 1999 conference. Three excellent panels were organized. This special issue of the *Journal of Systems and Software* includes extended version of some of the articles presented at the conference as well as several invited submissions.

While most everyone attending the conference agreed that proper software engineering education is a key in addressing many issues related to industrial software development, not everyone necessarily agreed on what the most proper approach should be. Some suggested a special track (or sub-area) on software engineering within a computer science program while others promoted an independent curriculum specifically designed to educate and train software engineers. Of course the proponents of each approach have their own reasons why their approach is most effective. One common point: industrial-relevant case studies and projects matter and should be an integral component of various options.

A summary of the included articles from the conference follows. Klaas Sikkel and his colleagues present a real-world case study for undergraduate students. Real-world case studies are different from textbook case studies and it is important to complement the academic skills and knowledge with some feeling of what real projects are about. The case in this course is the replacement of a Hospital Information System, a project that was recently carried out in a large regional hospital. An introductory program serves to prepare the students for the expected team skills and the background of the case study. The authors claim that the success of the course is based on three factors that mutually reinforce each other: training in management and communication skills, integration and application of knowledge gained at previous courses, and learning by experience the difference between a real-world problem and a textbook problem.

[*] Tel.: +1-402-554-2849; fax: +1-402-554-3284; e-mail: hossein@csalpha.unomaha.edu

[1] Guest editor.

0164-1212/99/$ - see front matter
PII: S0164-1212(99)00082-5

Claes Wohlin and Bjvrn Regnell present strategies for industrial relevance education for software engineers. They presents a collection of experiences related to success factors in graduate and postgraduate education. The experiences are mainly concerned with how to make the education relevant from an industrial viewpoint. This is emphasized as a key issue in software engineering education and research, as the main objective is to give the students a good basis for large-scale software development in an industrial environment. The presentation is divided into experiences at the graduate and postgraduate levels respectively. It is concluded that several challenging opportunities in making software engineering education relevant from an industrial perspective exist and that they are possible to implement in practice. The latter is based on the experiences reported in the paper.

Barrie Thompson and Helen M. Edwards discuss providing opportunities for new graduates. Their paper is concerned with developments and experiences associated with a postgraduate "conversion" course in computing that has operated at the University of Sunderland since 1989. Such conversion courses take graduates from other disciplines and provide them with a theoretical and practical understanding of a branch of computing. The Sunderland course is distinctive in that it has been offered in five different delivery modes that reflect flexibility in terms of attendance and responsiveness to the market in terms of curricula. Details are provided of the changing content of the course, the profiles of graduates on entry to the course, and thumbnail sketches of several graduates from the course. In particular, the approaches adopted towards the treatment of Software Engineering within the course are highlighted.

Michael Murphy presents a response-interaction approach to teaching software project management. According to Murphy, Southern Polytechnic State University has recently implemented a new Master of Science in Software Engineering degree, which includes a course in Software Project Management in its core requirements. Murphy's paper addresses an innovative approach to teaching this course through what is described as response-interaction. Also included are the results of the first offering of this course.

The rest of the articles are invited submissions and relate to the software engineering education, except the last one which introduces an Information Systems-Centric Curriculum. The last article included in the special issue because of its emphasis on collaborative and communication skills, large systems, as well as its promotion of a "system view" early in the program. These topics are quite relevant in a software engineering program too.

The paper by Vaughn and Boggess addresses an identity crises for the topic of computer security research and instruction and suggests that it belongs within a software engineering curriculum. Software engineering curricula today seldom address this important requirement, yet security engineering, tools, and techniques are often listed as National objectives and essential customer requirements. The authors believe that security is a user requirement and that it is achieved by good software engineering practices – to include formal specifications and verifications.

Kathy Beckman, Nancy Mead, and their colleagues elaborate on the activities of the SEI Working Group on Software Engineering Education and Training (WGSEE&T). The SEI Working Group has undertaken a multi-year study of industry/university collaborations. It first started to track industry/university collaborations in the Directory of Industry and University Collaborations with a Focus on Software Engineering Education and Training. Subsequently the Working Group undertook an in-depth study of this phenomenon. The study included a survey of collaborations, followed by interviews with the collaborations that seemed to have been most successful. The paper documents the study process, summarizes the survey results, and discusses the results of the follow-up interviews. More importantly, the paper documents the typical industry/university collaboration process and typical collaboration models and missions. The paper would be of interest to those who are engaged in such collaborations as well as those who would like to establish such a collaboration and want to know where to start.

Tom Hilburn et al. discuss issues and ideas for improving the undergraduate education of software engineers. They describes the work and results of a project to create a set of Guidelines for Software Engineering Education. The Guidelines address key problems in software engineering education and provide guidance and support for faculty trying to educate their students in software engineering. In its final form, it is envisioned that the Guidelines will contain the following: a description of the body of knowledge for software engineering (appropriate for curriculum development); a conceptual curriculum model; curriculum units and courses; sample curriculum implementations; curriculum support material; and guidance for assessment and accreditation.

Finally, Doris Lidtke and Gordon Stokes present a curriculum designed to prepare student to work with large systems, to develop collaborative and communication skills, and to take a system view from the very beginning. Their proposed curriculum, referred to as an Information Systems-Centric Curriculum, explores student learning and faculty teaching paradigms, and as claimed in the paper, provides an educational experience that produces students prepared to meet the increasing needs of industry to create and use complex information systems.

Preparation of this special issue has been an enjoyment. I would like to thank the authors for submitting their work. Working with the CSEE&T'99 Program Chair, Professor Don Bagert (Texas Tech), and the Program Committee has been equally an enjoyable process. I hope to see you all during the CSEE&T 2000 in Austin, Texas (March 2000).

References

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

Gibbs, W., 1994]. Software's chronic crisis. Scientific American 271 (3), 86–95.