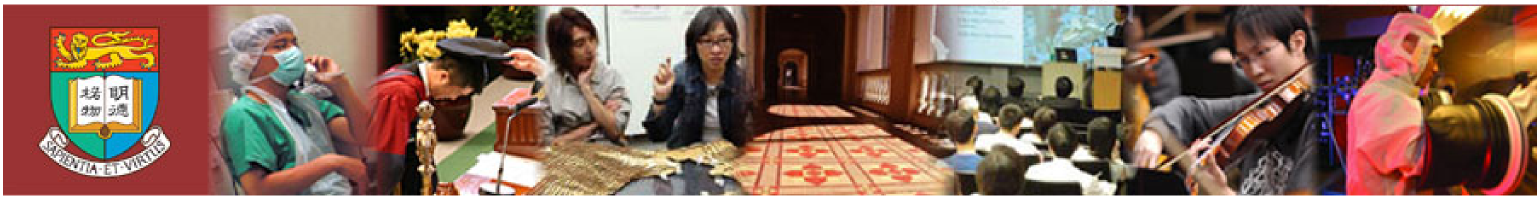

Computing Curriculum — Software Engineering: Its Impacts on Professional Software Engineering Education [*]

T. H. Tse
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
thtse@hku.hk

1. Introduction

The Software Engineering volume by the IEEE Computer Society / ACM Joint Task Force on Computing Curricula, otherwise known as the Computing Curriculum — Software Engineering (CCSE), has been finalized and approved [1]. This position paper comments on the impacts of the volume on professional software engineering education.

2. Educating Undergraduates to Become Professional Software Engineers

To enable undergraduates to become professional software engineers, we should note the three essential traits of professionalism commonly recognized [6]:

(*a*) A body of knowledge unique to the profession.

(*b*) High levels of responsibility and accountability via a code of ethics and professional practice.

(*c*) Competence and commitment in the profession demonstrated by certification by a professional organization or a legally recognized license to practise.

2.1. Body of knowledge

A few years back, I expressed my concern regarding the confusion between the curriculum for software engineering and those for other disciplines such as computing and engineering [5, 6]. The CCSE Final Report proves to be an excellent and comprehensive curriculum document specifying a body of knowledge for software engineers. On one hand, the present proposal considers SE as a discipline evolved from both computing and engineering. On the other hand, the SE curriculum is designed as a stand-alone mature programme unique in itself.

The curriculum is built up from 35% of computing essentials, 18% of mathematical and engineering fundamentals and 7% of professional practice. The main software engineering component comprehensively covers all the essential elements required in our profession, including software modelling and analysis, software design, software verification and validation, software evolution, software process, software quality and software management. The curriculum has been clearly justified through well-reasoned principles and guidelines.

2.2. Code of ethics and professional practice

Although CCSE recommends that

> "Graduates of an undergraduate SE program must be able to ... design appropriate solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal, and economic concerns",

we note that professionalism only make up 4% of the entire software engineering curriculum. Even though it suggests that

> "By taking opportunities to discuss these issues throughout the curriculum, they will be come deeply entrenched",

this is more easily said than done. From surveys such as Towell and Thompson [4], a quarter of the respondents indicate that professional ethics is only covered in a single course of their curriculum. Professional practices used throughout the curriculum may only be limited to software engineering standards.

2.3. Certification and licensing

It is perfectly legitimate for CCSE to recommend software engineers to adhere to the guideline in the *Software*

[*] This work is supported in part by a grant of the Research Grants Council of Hong Kong, a grant of the Croucher Foundation and a grant of the University of Hong Kong.

0730-3157/04 $20.00 © 2004 **IEEE**

Engineering Code of Ethics and Professional Practice [2], that "software engineers must commit themselves to making software engineering a beneficial and respected profession." It would be impossible, however, to exercise this without certification or licensing.

We cannot blame CCSE alone. Software engineers are still divided among ourselves as to whether we should be licensed. A survey by Towell [3], for instance, shows that 44% of us are for licensing and 39% are against it. The maintenance of professional conduct can only be feasible in the presence of licensing. The wish to dodge legal consequences for unprofessional behaviour has in fact been quoted as one of the reasons for opposing the licensing of software engineers [6].

3. The Clientele

Like any other system, it pays to know more about our clients — the software engineering students.

CCSE is intended to support undergraduate software engineering education. At our University, we have been offering the first and only undergraduate SE programme in Hong Kong since 2000, based on a curriculum design very similar to CCSE. Our graduates are well received both by professional bodies and the industry.

After several years of experience, however, we find that:

(*a*) Undergraduates students do not fully appreciate the SE approach to software development, since most of them do not have experience working in huge software systems in large teams. Some of them follow the SE practices to satisfy the needs of the curriculum rather than being fully convinced by the philosophy.

(*b*) There is a bigger demand from students who have already gone through an undergraduate curriculum in another computing or engineering discipline, have some experience in software development, and wish to excel in software engineering. They have a better understand of the philosophy and structure of the curriculum, and see a better need for the SE practices.

(*c*) Course evaluations by undergraduate SE students show that they are more comfortable with technical computing courses than software engineering courses that involve methodologies, processes, and professionalism. On the other hand, the evaluations improve as the students mature, and the highest evaluations are received from MSc students. For example, when Unified Modelling Language and Unified Process are being taught, undergraduate students often question whether people in the industry really make use of such diagrams and processes, while experienced MSc students would welcome more coverage of methodologies.

To meet real-life demands, therefore, we are moving towards an MSc programme in software engineering. Basic computing fundamental will not need to be covered again for these postgraduate students. Instead, because of the maturity and experience of the students, we can embark directly on to the software engineering elements of the programme. Other advanced computing, mathematical and engineering elements can be taught as supporting courses.

4. Conclusion

To conclude, CCSE is an excellent curriculum document that defines the body of knowledge for undergraduate software engineering students. If it can be coupled with carefully planned training on professionalism and ethics, and proper recognition of the graduates through accreditation and licensing, then it will definitely have a positive impact on the education of software engineers. Considerations should also be made to the level of students that will benefit most from SE education.

References

[1] Computing Curriculum — Software Engineering: Final Report. IEEE Computer Society / ACM Joint Task Force on Computing Curricula, 2004. Available at "http://sites.computer.org/ccse/volume/FinalReport-5-21-04.pdf".

[2] Software Engineering Code of Ethics and Professional Practice. ACM / IEEE Computer Society Joint Task Force on Software Engineering Ethics and Professional Practices, 1999. Available at "http://www.acm.org/serving/se/code.htm".

[3] E. Towell. Teaching ethics in the software engineering curriculum. In *Proceedings of the 16th Conference on Software Engineering Education and Training* (*CSEE&T 2003*), pages 150–157. IEEE Computer Society Press, Los Alamitos, California, 2003.

[4] E. Towell and J. B. Thompson. A further exploration of teaching ethics in the software engineering curriculum. In *Proceedings of the 17th Conference on Software Engineering Education and Training* (*CSEE&T 2004*), pages 39–44. IEEE Computer Society Press, Los Alamitos, California, 2004.

[5] T. H. Tse. Towards harmonized professional standards for software engineers: constraints, conflicts and concessions. In *Proceedings of the 24th Annual International Computer Software and Applications Conference* (*COMPSAC 2000*), pages 346–347. IEEE Computer Society Press, Los Alamitos, California, 2000.

[6] T. H. Tse. Software engineering professionalism: is the end of constraints and conflicts in sight? In *Proceedings of the 26th Annual International Computer Software and Applications Conference* (*COMPSAC 2002*), pages 347–348. IEEE Computer Society Press, Los Alamitos, California, 2002.

0730-3157/04 $20.00 © 2004 **IEEE**