

Towards Harmonized Professional Standards for Software Engineers: Constraints, Conflicts and Concessions *

T. H. Tse

Department of Computer Science and Information Systems
The University of Hong Kong
Pokfulam Road, Hong Kong
Email: tse@csis.hku.hk

1. Introduction

The harmonization of professional standards usually means an attempt to unify the standards among different nations or states. It is a necessary step towards the maturity of a profession because of two factors: (a) that professional standards have been developed independently in different nations, and (b) that the standards thus developed are not uniform among nations.

To test whether we have reached this mature stage in the software engineering profession, let us consider the following multiple-choice question:

Who are permitted to practice as software engineers?

(A) Graduates in software engineering; (B) Graduates in computer science; (C) Programmers, operators, and so on, who have been in the field for many years; (D) People who have attended short courses by Microsoft or similar vendors; (E) Graduates or non-graduates from related or unrelated disciplines who are smart enough.

The answer for most nations, unfortunately, is "all of the above". In other words, the software engineering profession has not reached a mature stage where harmonization is imminent.

On the other hand, the ACM Council has just decided on the other extreme [1]. They have voted to withdraw from the joint Software Engineering Coordinating Committee (SWECC) that have been trying to address the licensing issue very seriously.

Since harmonization is not the only step before a mature software engineering profession, we propose, instead, to take a look at various constraints and conflicts against harmonized professional standards in software engineering.

*This research is supported in part by the Hong Kong Research Grants Council and the University of Hong Kong Committee on Research and Conference Grants.

2. Constraints and Conflicts

2.1. Body of knowledge

It has generally been observed that "Many engineers and computer scientists are ... concerned about the validity of software engineering as an engineering discipline" [2]. For example, there is a lot of confusion between software engineering and computer science. Consider the survey conducted by the IEEE Computer Society / ACM Joint Task Force on Body of Software Engineering Knowledge. The following are listed as the "tasks expected of novices at different rates of endorsement" [3]:

Algorithm complexity, caches, client server, computer peripherals, data management, data models for databases, database administration, database performance and capacity planning, database system fundamentals, database systems, data structures, device drivers, distributed systems, effort estimation, kernels, project management and planning, power management, real time systems, software quality assurance, static/dynamic linking, and transaction properties

Similar items are listed by the Texas Board of Professional Engineers. If such confusion between SE and CS exists in the leading bodies who advocate the professionalism of software engineering, it is hardly surprising that more confusion exists among other engineers and computer scientists.

A position statement issued by the ACM Council [1] says that "Licensing as Professional Engineers would be impractical for software engineers, because it would require examinations over subjects most software engineers neither study in their formal education nor need in order to practice competent software engineering." This further illustrates the fallacious view that the body of knowledge of our profession is computer science, even though ACM also have

a contradicting view that such knowledge is unnecessary for software engineers to practice competently.

2.2. Professional ethics and public interest

Numerous discussions have been made on issues related to the licensing of software engineers. Have they been made in the interest of the general public, or the interest of those who wish to have a piece of the pie in the system?

For example, the licensing of software engineers has been met with strong resistance in some engineering circles. They challenge whether software engineers are indeed engineers in the traditional sense. A licensing system is also challenged by people who may not have the proposed background knowledge, but are already actively practising the trade. They are concerned whether they will be allowed to continue in the new system. We must emphasize that any licensing system is a measure to maintain the professional standards, rather than being used as a barrier of entry into the profession. For example, it would be impossible to seriously address the ethics of professional practice in the absence of a licensing system.

On the other hand, the general public is more interested in their own benefits resulting from the licensing of software engineers. In simple terms, will the quality of software improve? Will the software projects meet the deadlines by appointing licensed professionals? Will software engineers be able to sign above the dotted line certifying the safety of the software produced? Such issues must be addressed properly.

2.3. Asia concern

International experts are often invited to sit on the review boards that help to accredit degree programmes for professional institutions in Asia. In general, international experts are objective and their advice is very useful. However, we have encountered many occasions where the expert opinion may be against the interest of software engineering profession.

Experts in developed countries may have different points of view on different issues and have not yet reached a consensus. When they are asked to give advice to “developing” countries, however, they often portray their personal opinion as the orthodox view. Sometimes they may even try to regain lost ground by imposing their outdated view in “developing” countries. For example, I have personally witnessed several occasions where “sufficient hardware engineering elements” are set as a requirement in software engineering programmes. This is certainly not the kind of harmonization that we are looking for.

3. Concessions and Conclusion

As suggested by the ACM position statement [1], the software engineering profession has not reached a mature

stage of development. On the other hand, I disagree with their position that our state of knowledge and practice in software engineering is too immature to warrant licensing.

The software engineering profession may not be immediately ready for licensing and harmonization, but we must start the groundwork immediately. After all, “... the software community had better figure out how to do it before someone else does it for them.” [5]

Measures are required to help the young profession to mature. For example,

- (a) Although there is an urgent need for the definition of a body of knowledge in software engineering, such BoK should not be hastily copied and pasted from computer science or other engineering disciplines. The BoK may contain topics from related disciplines, but should not be confused with them.
- (b) Even though there is an ultimate need for harmonized professional standards, this should be achieved via a series of consultations and discussions, rather than by thrusting the standards of one discipline on another, or by imposing the standards of one nation on another.
- (c) There should be multiple tiers in the software engineering profession, differentiating professional engineers (who should be held responsible for the complete software engineering process and products) from other supporting technical staff such as programmers. We should concentrate on the licensing of professional software engineers as our initial objective. The licensing of support staff in software engineering may be considered as a second step.

References

- [1] ACM. A summary of the ACM position on software engineering as a licensed engineering profession. Final Version, July 17, 2000. Available at “http://www.acm.org/serving/se_policy/selep_main.html”.
- [2] D. Bagert. Licensing software engineers: the Computer Society must continue to lead. In *Proceedings of the 23th Annual International Computer Software and Applications Conference (COMPSAC '99)*, pages 200–201. IEEE Computer Society, Los Alamitos, California, 1997.
- [3] IEEE Computer Society/ACM Joint Task Force on Body of Software Engineering Knowledge. Report on analyses of pilot software engineer survey data. 1997. Available at “<http://www.computer.org/tab/seprof/survey.htm>”.
- [4] IEEE Computer Society/ACM Joint Task Force on Software Engineering Ethics and Professional Practices. Software engineering code of ethics and professional practice. Available at “<http://www.computer.org/tab/seprof/code.htm>”.
- [5] N.R. Mead. Issues in licensing and certification. In *Proceedings of the 10th Conference on Software Engineering Education and Training (CSEE '97)*, pages 150–160. IEEE Computer Society, Los Alamitos, California, 1997.