What do you mean I've been practicing without a license? Certification & Licensing of Requirements Engineering Professionals

Annie I. Anton
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
aianton@eos.ncsu.edu

Joanne M. Atlee
Department of Computer Science
University of Waterloo
Waterloo, ON N2L 3G1 Canada
jmatlee@se.uwaterloo.ca

Given the short supply of software professionals and the number of practitioners who are self-educated, there is growing concern about the qualifications of those entrusted to develop high quality, reliable software. One response has been a movement to license software professionals. In 1998, the Texas Board of Professional Engineers began licensing software engineers as professional engineers. Partly in response to requests from the Texas Board of Engineers, the IEEE Computer Society is establishing a set of guidelines for professional organizations that are interested in licensing software engineers [3]. The IEE and the British Computer Society in the UK, under the umbrella of the Engineering Council, have been registering software engineers as chartered engineers for more than 10 years.

This momentum towards licensing is a source of much controversy and concern among software developers. The November 1999 issue of IEEE Software, entitled "Professional Software Engineering: Fact or Fiction?" includes a wide range of opinions and positions on whether software engineering is ready to be a profession. The Association of Computing Machinery (ACM) Council has come out against "the licensing of software engineers at this time because the ACM believes that it is premature and would not be effective at addressing the problems of software quality and reliability" [1]. They are, though, willing to act as an advisor on the body of knowledge and best practices in software engineering for other organizations that are interested in licensing software professionals [1].

How should the licensing of software engineers affect those of us who work on requirements-related activities? Given that several studies have shown that the majority of software errors can be traced to missing, incorrect, or misunderstood requirements [2, 4, 5, 6], it seems especially appropriate to license those professionals who develop requirements for high quality and reliable systems. However, the techniques and processes used to formulate requirements may not be as mature or as repeatable as the practices used in other software engineering activities.

We would like the panel to address these questions:

- *What are the goals we want to achieve?*
  If software quality and reliability are the ultimate goals, is the licensing of software practitioners the best way to achieve them?
- *Who should be licensed?*
  In particular, should practitioners who work on requirements-related activities be licensed?
- *Who should be the licenser?*
  In North America, the professional engineering societies have taken the lead in licensing (or planning to license) software engineers. Should the ACM take a more active role?
- *Are our best requirements practices sufficiently mature and stable to be called engineering activities?*
  Can we agree on the requirements notations, processes and activities that belong in a core body of knowledge for software engineering?

References

[1] ACM panel on professional licensing in software engineering report to council, May 15 1999. http://www.acm.org/serving/se_policy/report.html.

[2] B. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.

[3] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, and L. Tripp.Robert. The guide to software engineering body of knowledge. *IEEE Software*, pages 35–44, November 1999.

[4] N. Leveson. Software safety: Why, what, and how. *Computing Surveys*, 18(2):125–163, June 1986.

[5] R. Lutz. Analyzing software requirements errors in safety-critical, embedded systems. Technical Report TR92-27, Department of Computer Science, Iowa State University, 1992.

[6] T. Nakajo and H. Kume. A case history analysis of software error cause-effect relationships. *IEEE Transactions on Software Engineering*, 17(8):830–838, August 1991.