

Issues in Licensing and Certification of Software Engineers¹

Prepared for the ACM/IEEE Steering Committee on Software Engineering as a Profession

**Nancy R. Mead
Software Engineering Institute
4500 5th Avenue
Pittsburgh, PA 15213-3890
Phone: (412) 268-5756
Fax: (412) 268-5758
EMail: nrm@sei.cmu.edu**

¹ This work is sponsored by the U.S. Department of Defense.

Issues in Licensing and Certification of Software Engineers

Abstract

The issues associated with licensing and certification of software engineers are difficult. At present, there is no agreed-to body of knowledge on which to base certification. Some state legislatures are attempting to regulate the practice of software engineering without adequate understanding of the field. As a result of safety-critical software disasters, some professionals believe that licensing or certification is inevitable, so the software community had better figure out how to do it before someone else does it for them. In this paper, we survey the state of the practice of licensing and certification in other professions, identify the issues that might be encountered in attempting to license and certify software engineers, and suggest possible actions that could be taken by the ACM/IEEE Steering Committee.

1. Introduction

The ACM/IEEE Steering Committee for the Establishment of Software Engineering as a Profession has started three task forces in the areas of "Software Engineering Body of Knowledge and Recommended Practices," "Software Engineering Ethics and Professional Practices," and "Software Engineering Curriculum." As part of the overall steering committee's activities, a white paper was requested on the subject of issues in licensing and certification of software engineers. This paper briefly discusses the work of the steering committee, the history of licensing and certification in other disciplines, and current certification bodies in areas related to software engineering. We identify issues associated with licensing and certification in software engineering, and possible courses of action for the steering committee and others concerned with this issue.

2. Background

In May 1993, the IEEE Computer Society Board of Governors approved a motion to establish a steering committee to explore actions related to establishment of software engineering as a profession [Barbacci 1993]. In November of 1993, the steering committee, chaired by Mario Barbacci, made its initial report to the Board of Governors. The steering committee made four recommendations:

Adopt standard definitions

Define a required body of knowledge and recommended practices

Define ethical standards

Define educational curricula

A similar set of actions was endorsed by the ACM Council in August 1993, with the stated desire to work jointly with the IEEE Computer Society. Specifically, the ACM chartered a Commission on Software Engineering to provide a white

paper that assessed software engineering relative to the following questions [Zweben 93]:

1. *What activities are considered part of software engineering? How is design currently formulated with software engineering? What other formulations of software design are in use? How can they be brought together?*
2. *What is a profession? What is the state of software engineering as a profession?*
3. *What is a discipline? What is the state of software engineering as a discipline?*
4. *What standard practices currently exist for software engineering? How do they compare with those in other fields recognized as disciplines or professions?*
5. *What responsibilities are generally expected of persons with the title of "software engineer"? "Designer"? "Architect"? Are there useful distinctions in the responsibilities of persons with related job titles?*
6. *What are the important issues to be confronted within the various subareas of software engineering?*
7. *What are the implications of the above for education? Curriculum? Accreditation?*
8. *What are the implications of the above for certification? Licensing?*
9. *What actions are recommended for ACM?*

In February 1994, the first four IEEE recommendations were documented in the "Standards" column of *IEEE Computer* [Barbacci 94a]. In March 1994, the report to the IEEE Computer Society Board of Governors indicated that the steering committee would be structured to be a joint ACM/IEEE Steering Committee for the Establishment of Software Engineering as a Profession [Barbacci 94b]. Stu Zweben was named as the vice-chair, and stepped down when he became ACM President. Zweben was replaced as vice-chair by Dennis Frailey. The two society presidents at that time, Laurel Kaleda and Stu Zweben, were named as ex-officio members [Boehm 94]. The steering committee recommended establishment of the three task forces: Define the Body of Knowledge and

Recommended Practices, chaired by Pat Douglas of IBM; Define the Code of Ethics and Professional Standards, co-chaired by Robert Melford and Don Gotterbarn; and Define the Curriculum, co-chaired by Doris Carver and John Werth. In late 1995, Barbacci stepped down as chair of the Steering Committee, since he was President-elect of the Computer Society, and was replaced as chair by Felipe Cabrera.

The Define the Body of Knowledge and Recommended Practices Task Force developed a survey instrument, which is being prototyped and distributed in 1996 [Frailey 1996]. Funding for distribution and analysis was provided by the U.S. Army, and volunteers are performing an in-depth review. The survey consumed much of the effort and energies of the task force and steering committee in 1995. The Define the Code of Ethics and Professional Standards Task Force has established working groups in the areas of privacy, reliability and safety, security, social justice, institutional support, intellectual property, professional competence, and professional relationships. The Define the Curriculum Task Force leadership has been appointed by the steering committee and is working with the Define the Body of Knowledge and Recommended Practices Task Force. Much of its work depends on the survey results.

In early 1996, it was decided that some consideration might be given to licensing and certification issues. This resulted in actions to characterize the different kinds of certification and produce a white paper on the issue of credentials. Mary Shaw produced an initial taxonomy in February 1996, and this white paper addresses credentials (licensing and certification) issues.

Shortly after these actions were assigned, the National Software Council decided to hold a workshop in June 1996 on licensing and certification of software engineers. This created a natural forum for presentation of this paper.

3. Licensing and Certification

3.1 Definition

The terms *licensing*, *certification*, and *registration* are used in various ways, sometimes interchangeably and sometimes overlapping. In order to communicate on these issues, it helps to have some common definitions. Ford's report [Ford 96] gives some informal definitions of certification and licensing which seem useful in the context of this discussion:

Certification is a voluntary process administered by a profession.

Licensing is a mandatory process administered by a governmental authority.

3.2 Licensing and Certification in Other Professions

Licensing and certification have been performed in other professions for years. To have a context for licensing and certification of software engineers, it helps to understand licensing and certification in other professions. All of this information is fairly well-known and has been documented many times, so we will only provide a brief summary here.

Ford [Ford 96] discusses professional certification programs for other professions, such as accounting, medicine, and engineering. He also refers to the ICCP, discussed below, and the American Society for Quality Control (ASQC). In her taxonomy, Shaw [Shaw 96] makes reference to the medical profession.

The Certified Public Accountant (CPA) designation is well known in the accounting profession. In medicine, the National Board of Medical Specialties offers certification in more than 20 areas of medicine. The National Institute for Certification in Engineering Technologies provides certification for five engineering support positions: Associate Engineering Technician, Engineering Technician, Senior Engineering Technician, Associate Engineering Technologist, and Certified Engineering Technologist [Ford 96].

Certification tends to be done by professional societies and non-profit organizations. Commercial companies, discussed below, also certify users of their products and services.

Licensing, on the other hand, is generally administered at the state level in the United States. Protection of the public is usually the motivation for licensing. Medicine and law are the licensed professions most widely known to the public, but engineers are also licensed. The percentage of engineers in common engineering disciplines who seek licenses is shown in Table 1 [Ford 96].

Table 1. Percentage of Licensed Engineering Graduates

Discipline	Licensed
Civil	44%
Mechanical	23%
Electrical	9%
Chemical	8%
All Engineers	18%

In discussing the legal aspects of licensing, Ford quotes from section 6747 of the California business and professions code:

This chapter, except for those provisions which apply to civil engineers and civil engineering, shall not be applicable to the performance of engineering work by manufacturing, mining, public utility, research and development or other industrial corporation or by employees of such corporation, provided such work is in connection with or incidental to the products, systems, or services of such corporation or its affiliates.

Most engineers are not licensed. The safety considerations associated with civil engineering are the probable reason that many civil engineers are licensed.

Shaw suggests that registration of software engineers is analogous to registration in medicine. In medicine, there are two levels of credentials: the overall MD degree combined with the state license, and board certification in

specialties. Shaw categorizes these forms of registration as knowledge-intensive and relatively independent of specific technologies.

4. Current Certification Bodies in Areas Related to Software

The most well-known certification organization in the software area is the Institute for Certification of Computing Professionals (ICCP). There are two levels of certification, certification as an Associate Computing Professional or as a Certified Computing Professional. The Associate Computing Professional (ACP) certification requires successful completion of exams on core topics and one language exam. Language exams were traditionally offered in FORTRAN, Pascal, BASIC, RPG II, RPG/400, COBOL, C, and Ada. Naturally, the list of languages has been subject to revision, so that, for example, C++ is now an accepted language for certification. On the other hand, FORTRAN, RPG II, and Ada were dropped in 1994.

The Certified Computing Professional (CCP) certification requires successful completion of exams on core topics and two among the following topics: Management, Procedural Programming, Business Information Systems, Communications, Office Information Systems, Systems Security, Software Engineering, Systems Programming, and Data Resource Management. In addition, the CCP candidate must have the equivalent of 48 months of full-time experience in information systems, or academic credentials with 24 months of full-time experience. For example, a bachelor's degree in information systems or computer science is considered the equivalent of 24 months experience. All candidates must subscribe to a specified Code of Ethics, Conduct, and Good Practice. The ICCP certification procedures and exams are revisited at least annually and can be expected to change periodically.

In 1995, ICCP and Learning Tree International announced a strategic alliance that would allow Learning Tree exams to be accepted as partial fulfillment of the ACP and CCP requirements. For example, the Learning Tree exam in C++ can be substituted for one language exam when a candidate applies for CCP or ACP certification. Further, all Learning Tree courses will be accepted by ICCP toward the CCP continuing education requirements.

The American Society for Quality Control (ASQC) has a certification process for a Software Quality Engineer. The ASQC certification requirements include:

Eight years of professional experience. At least three years must be in a decision-making position. A bachelor's degree may be counted as four years of experience, or an advanced degree may be counted as five years of experience.

Proof of professionalism. This may be membership in an appropriate professional society, holding a professional engineer's license, or statements from two professional colleagues.

Completion of a written examination. The written exam is in the areas of software quality management, software engineering, project management, appraisal, issues, analytical methods, and quality systems [Ford 96].

Like the ICCP, the ASQC has a code of ethics.

Many commercial software companies certify practitioners in the use of specific tools; for example, Apple, Microsoft, and Novell. This form of certification is discussed in Ford's report [Ford 96] and Shaw's paper [Shaw 96]. Shaw characterizes this form of certification as approval by a manufacturer for an individual to represent themselves as skilled in the use of the manufacturer's product. This is not the type of certification that the professional societies need to be concerned with, except to ensure that manufacturer's certifications are not confused with other kinds of credentials. Ford characterizes them as certification programs offered by commercial companies in their own products and services.

5. Current Licensing Bodies in Areas Related to Software

Recently it was announced that a Professional Engineer exam for "Computer Engineer" would be developed by the National Council of Engineering Examiners and Surveyors (NCEES). This does not seem to have any direct bearing on the issue of licensing of software engineers, although it certainly suggests increased attention to licensing of computing professionals in general.

6. Issues

Issues that must be resolved before establishing certification of software engineers include the following:

- 1) Will software engineers use the "software engineer" title, or will another title be used to describe the certified professional? If the term software engineer continues to be used, will there be backlash from the licensed Professional Engineers?
- 2) What body will perform certification? The front-runner among the existing certification bodies would seem to be ICCP, perhaps in combination with the ACM and IEEE Computer Society.
- 3) What is the body of knowledge against which software professionals are to be certified?
- 4) Are all software engineers to be certified or only those engaged in the design of safety-critical or trustworthy systems?
- 5) Within those specialties, is it necessary to certify all practitioners, or is it sufficient to have one certified software engineer who is responsible for the software system?

Among the licensing issues are these:

- 1) How will licensing exams be constructed? If software engineers are licensed as Professional Engineers, it seems likely that NCEES will construct and administer the exams.
- 2) Will licenses carry over from one state to another?
- 3) Is licensing an idea that makes sense for some software specialties but not others?

4) What about use of the term engineer? What are the legal issues?

7. Possible Courses of Action

We could push for licensing or certification, or we could continue to take no action at all. To propose a course of action, we need to determine what is needed as a prerequisite to licensing or certification. There are many opinions on this; two examples are included below.

1. Shaw [Shaw 96] suggests that essential elements to certification are:

1. *An achievable level of practice that provides reasonable assurance of safety, efficacy, utility, and other public interest concerns*
2. *A test instrument that can discern whether an individual is capable of practicing at that level and likely to do so.*

Shaw points out that we have not yet identified the first of these elements. In her view the role of the traditional university degree is an orthogonal issue, and might or might not provide good preparation for practice of a specialty.

2. Frailey [Frailey 96] suggests that the requirements for software registration include:

- Established body of knowledge*
- Established practices ("building codes")*
- Accredited education programs*
- Established codes of professional conduct and ethics*
- Examinations*

In Frailey's opinion, we do not have an accepted body of knowledge, education program, or licensing system; nor do we have established practices.

There are several possible courses of action that range from doing nothing to making an all-out effort to get certification of software engineers in place. A middle ground might be to start a modest effort that would allow the profession to be prepared to do certification when the time is right.

8. Recommendations

Some would argue that any discussion of licensing or certification is premature, in the absence of an accepted body of knowledge. It is our opinion that the software profession needs to prepare for eventual licensing or certification. This activity could start with a task force chartered by the ACM/IEEE Steering Committee to examine the various alternatives in more depth and make recommendations to the Committee. In the near term, it would be useful to identify potential certification bodies. Then it might make sense to consider which specialties could be subject to licensing, or whether all software engineers are likely to need to be licensed. By the time this groundwork is laid, further information about the body of knowledge should have become available, and we will have a better sense of whether licensing or certification will eventually take place. Using this approach, software engineers will be able to have a more proactive role, rather than leaving it to legislators with little or no background in the field, and then trying to react or undo inappropriate legislation after the damage is done.

References

- [Barbacci 93] Aylor, J.; Barbacci, M.; Buckley, F.; Druffel, L.; Royce, W.; Schneidewind, N.; Zweben, S. *Report to the Board of Governors of the IEEE Computer Society from the Steering Committee for the Establishment of Software Engineering as a Profession.* November 10, 1993.
- [Barbacci 94a] Barbacci, M. "Panel presents first four recommendations aimed at establishing software engineering as a profession." *IEEE Computer,* (February 1994): 80-81.
- [Barbacci 94b] Barbacci, M.; Buckley, F.; Chikofsky, E.; Druffel, L.; Royce, W.; Schneidewind, N.; Zweben, S. *Report to the Board of Governors of the IEEE Computer Society from the Steering Committee for the Establishment of Software Engineering as a Profession.* San Francisco, March 4, 1994.
- [Boehm 94] Boehm, Barry. "The IEEE-ACM Initiative on Software Engineering As A Profession," *IEEE Computer Society Software Engineering Technical Council Newsletter* 13, 1 (September 1994): 1-2.
- [Ford 96] Ford, Gary & Gibbs, Norman. *A Mature Profession of Software Engineering.* (CMU/SEI-96-TR-004). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1996.
- [Frailey 96] Frailey, Dennis J., "Issues in Licensing of Software Engineers as Professionals," Dallas/Fort Worth SPIN Meeting, February 6, 1996.
- [Shaw 96] Shaw, M. "An Initial Certification for Taxonomy," Working Paper, Carnegie Mellon University, January 23, 1996.
- [Zweben 93] Aylor, J.; Barbacci, M.; Buckley, F.; Druffel, L.; Royce, W.; Schneidewind, N.; Zweben, S. *Report to the Board of Governors of the IEEE Computer Society from the Steering Committee for the Establishment of Software Engineering as a Profession.* Appendix 2, November 10, 1993: 8-10.