

# **The Emergence of Software Engineering Professionalism**

## **The Role of Professional Societies in the Emergence of Software Engineering Professionalism in the United States and Canada**

**Stephen B. Seidman**

University of Central Arkansas, USA, sseidman@uca.edu

**Abstract:** The term “software engineering” was coined forty years ago as a metaphor for the processes involved in designing and constructing large-scale software systems. Since that time, the term has gradually come to refer to a professional engineering discipline. This paper will describe the role played by professional societies in the creation of the professional discipline of software engineering in the United States and Canada.

**Keywords:** software engineering, professionalism, professional societies, United States, Canada

### **1. Software Engineering as an Engineering Profession**

The first recorded use of the term “software engineering” was at a NATO conference in 1968. According to the conference organizers [1], the term was chosen “was deliberately chosen as being provocative, in implying the need for software manufacture to be [based] on the types of theoretical foundations and practical disciplines[,] that are traditional in the established branches of engineering”. This concern arose out of a growing realization that the software development practices used at that time were hampering the development of industrial software systems and applications [2].

In 1968, the “engineering” label was used metaphorically; it was applied to software in the hope that engineering principles could be applied to software design and development. At one level, the success of the label is clear; “software engineering” quickly came to refer to the processes associated with software development. For example, the most prestigious conference in the software engineering discipline is the International Conference on Software Engineering,

which traces its roots to a US conference held in 1975. However, it is less clear that applying the “engineering” label to software has led to the transfer of the attributes that characterize the engineering disciplines. Mahoney [2] argues that those who coined the phrase “software engineering” were not completely clear on the specific principles that they thought should be adopted by software developers. Engineering principles mentioned in the early days of software engineering include assembly of systems from components and Taylor-style industrial engineering.

However, engineering is more than a collection of principles. Just as medicine, law, and architecture are professions, engineering is also a profession, and an engineer is therefore a professional. The concept of “professionalism” cuts across all professions. According to Sandra Day O’Connor, retired Associate Justice of the US Supreme Court, “... the essence of professionalism is a commitment to develop one’s skills to the fullest and to apply that responsibly to the problems at hand. Professionalism requires adherence to the highest ethical standards of conduct and a willingness to subordinate narrow self-interest in pursuit of the more fundamental goal of public service” [3].

What are the characteristics of a profession? First, the community of practicing professionals must have responsibility for governance of the profession. For this purpose, the professional community is typically represented by one or more self-governing professional bodies. The community is also responsible for controlling entrance to the profession and establishing and enforcing standards of professional behavior.

How does this apply to engineering? Professional engineering bodies can be organized in two ways. First, a society may be responsible for a specific engineering specialty. A typical example is the Institute of Electrical and Electronic Engineers (IEEE). Specialty societies are often organized into an umbrella body, such as the Engineering Institute of Canada. A second form of organization derives from the federal political structure of some countries. In the United States and Canada, for example, individual states and provinces have engineering societies that cut across specialties, such as the Texas Society of Professional Engineers and Professional Engineers Ontario. These local societies are also organized into umbrella bodies: the National Society of Professional Engineers in the US, and the Canadian Council of Professional Engineers (also known as Engineers Canada) in that country.

These professional engineering bodies exercise control over the profession in cross-cutting ways. For example, in the United States, access to the engineering profession is controlled in two distinct ways. First, the 50 states award professional licenses to engineers; licensure is analogous to “chartered status” in the United Kingdom. The criteria for licensure include passing examinations written and administered by the National Society of Professional Engineers. Another dimension relates to the educational credentials of engineers; licensure requirements are lower for holders of bachelor’s degrees from accredited university programs. Engineering accreditation in the US is administered by

ABET (formerly known as the Accreditation Board for Engineering and Technology); the members of ABET are the specialty societies mentioned above.

The specialty societies have a major influence on the professional conduct of their members. One key element of professionalism is the ethical component of professional behavior. In order to foster this component, each specialty engineering society has developed a code of ethics that is incumbent on its members. For example, the first of ten points in the IEEE code [4] requires IEEE members to “accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment”.

How does software engineering fit into this professional engineering paradigm? This question can be answered in two ways. First, has software engineering moved beyond its metaphorical roots and come to resemble the other engineering disciplines? Second, has software engineering developed the organizational characteristics of an engineering profession?

Mary Shaw [5] has addressed the first question. She argues that engineering practice is characterized by phrases like “creating cost-effective solutions”, “to practical problems”, and “by applying scientific knowledge. Shaw describes the emergence of civil and chemical engineering in the nineteenth century, and uses this history to discuss the position of software engineering on the “path toward engineering”. Another perspective on the emergence of software engineering as a profession is given by Adams [6]; this paper compares the emergence of this profession in Canada, the US, and the UK.

The second question can be divided into several: How is entrance to software engineering controlled by professional bodies? What about professional licensure and program accreditation? Is there a body of knowledge for software engineering? Is there a code of professional ethics for software engineers? The following sections of this paper will discuss the ways in which professional bodies have addressed these issues in the North American context.

## **2. Software Engineering and Professional Bodies in the US and Canada**

### ***2.1 Professional Bodies and Academic Programs***

The United States is the home of two major professional computing societies: the IEEE Computer Society (IEEE-CS) and the Association for Computing Machinery (ACM). Each society has more than 50,000 members; although both societies regard themselves as international, most of their members are in North America. The primary professional computing society in Canada is the Canadian Information Processing Society (CIPS).

For many years, IEEE-CS and ACM have worked together to develop recommendations for computing curricula. Recommendations have been produced for computer science, computer engineering, and information systems. In 2004, the two societies produced curriculum recommendations for software engineering (SE2004) [7].

In the United States, accreditation of university engineering programs was introduced by the Accreditation Board for Engineering and Technology in the 1930s. Computer science programs arrived on the academic stage much more recently. They were first accredited in the 1980s by the Computer Science Accreditation Board. The initial members of this board were the two computing professional societies: ACM and IEEE-CS. In the late 1990s, the engineering and computing accreditation bodies were renamed as their acronyms (ABET, CSAB). CSAB then joined ABET as a professional society, thus giving ABET responsibility for computing accreditation. ABET now accredits programs in a wide range of computing disciplines: computer science, computer engineering, information systems, information technology, and software engineering.

By contrast, the Canadian engineering and computing accreditation bodies are still separate. The Canadian Engineering Accreditation Board (CEAB) has responsibility for engineering accreditation, and CIPS has established the Computer Science Accreditation Council, with responsibility for accreditation of computing programs. While in the US, ABET and CSAB share responsibility for software engineering accreditation, CEAB claims that it has sole responsibility for accreditation of software engineering programs in Canada. This distinction is underscored by the fact that Engineers Canada has used the fact that it has registered the terms “engineer” and “engineering” as its trademarks as a basis for legal action to restrict academic programs in software engineering to departments and colleges of engineering. In this context, it has proved difficult for CEAB and CIPS to work together on software engineering accreditation, and both organizations now independently accredit software engineering programs [8].

## ***2.2 Professional Societies and Engineering Licensure***

In the United States and Canada, engineering licensure is a government-recognized professional status analogous to “chartered engineer” status in the United Kingdom and Australia. In North America, practicing the profession of engineering requires a license. However, “practicing the profession of engineering” has come to refer to the responsibilities typically associated with managing major projects. Such licenses are awarded locally, by states in the United States, and by provinces in Canada. In the US, each state has established its own regulations and requirements for engineering licensure, although the educational and experience requirements are similar. In all US states, candidates must pass two examinations common to all states: Fundamentals of Engineering (FE), and Principles and Practice (PE). The first examination covers knowledge

common to the traditional engineering disciplines (e.g., statics, mechanics, thermodynamics), while the second examination deals with advanced material specific to a particular engineering discipline (e.g. civil engineering, electrical engineering). In Canada, the educational requirements for licensing are met by graduation from an accredited engineering program.

If software engineering is an engineering discipline alongside other engineering disciplines, it would be reasonable to license software engineers, especially those working on systems that have critical health or safety implications. This has already happened in Canada, where the provinces of Ontario, Alberta, and British Columbia license software engineers. Progress has been much slower in the United States, where Texas is the only state that licenses software engineers. The primary reason for slow adoption in the US is the need to extend the licensure examination to correspond to the education of software engineers. Bagert [9] gives the context and history of software engineering licensing in the United States.

The role of professional societies is also relevant to the emergence of software engineering licensing. In the United States, the two professional computing societies have taken different positions on this issue. In general, IEEE-CS has taken a “watchful waiting” attitude. Since the parent organization IEEE is an international professional society, US-specific issues are the province of a subsidiary organization, called IEEE-USA. This latter organization has recently taken up the issue of software engineering licensing, and IEEE-CS is playing a cooperative but not a leading role in these discussions.

By contrast, ACM’s leadership body (ACM Council) adopted a position in 2000 that expressed strong opposition to software engineering licensing. This opposition was based on the contention that the software engineering discipline was not yet sufficiently mature [10]. One consequence of this decision was ACM’s withdrawal from cooperation with IEEE-CS on matters relating to software engineering professionalism.

### ***2.3 Professional Societies and Certification***

It is important to distinguish between broad-based certifications and product-specific certifications. Broad-based certifications are based on bodies of knowledge that cover an entire professional discipline or a subspecialty within such a discipline. These certifications are generally awarded by professional societies. Examples of broad-based certifications include financial certifications and specialty certifications in medicine or law. In the computing domain, broad-based certifications are available for software engineers and security experts. By contrast, product-specific certifications are based on a specific product or product line, such as a medical device or an operating system. The manufacturers of the products or product lines usually award certifications tied to their products.

Candidates applying for a broad-based certification must meet specific education and experience requirements. A candidate’s familiarity with a body of knowledge is generally assessed by examination, although some certification

schemes use peer review to assess knowledge and/or professional experience. Most certification schemes require that a certificate holder demonstrate professional activity, commitment to a code of ethics, and continuing education in order to maintain certification. Broad-based certification schemes are governed by national and international standards.

The IEEE Computer Society's Certified Software Development Professional (CSDP) certification scheme is an example of a broad-based software engineering certification. It is targeted to professionals with four years of professional software engineering experience. The origin of the CSDP scheme can be traced back for almost a decade. In 1998, the IEEE Computer Society began to consider the feasibility of certifying software engineering professionals. The CSDP examination was developed over the following three years, with its first offering in 2002. It consists of 180 questions to be completed in 3.5 hours. The examination is offered at testing centers in many countries. There are currently more than 600 CSDP certificate holders, who reside in many countries in all parts of the world. The IEEE Computer Society's Professional Practices Committee (PPC) is currently revising the examination to bring its body of knowledge into conformance with the revision of the SWEBOK body of knowledge (see below). At the same time, the PPC is developing a new examination targeting recent university graduates. This certification will be called the Certified Software Development Associate (CSDA).

An international standard for software engineering certification schemes is currently under development by an ISO/IEC JTC1 working group. An extended discussion of professional certification in software engineering is given in [11].

#### ***2.4 Professional Societies and Standards***

Another aspect of professionalism is the establishment of standards for professional practice. One way of assessing the state of software engineering professionalism and the role played by professional societies is to examine the world of software engineering standards. In general, technology standards are developed and managed at many levels: by professional and industry organizations, by national standards bodies, and by international organizations.

IEEE is an excellent example of a professional organization with a responsibility for standards development across a wide range of technical specialties (see [www.standards.ieee.org](http://www.standards.ieee.org)). Within this sphere of activity, the IEEE Computer Society has been actively involved in developing software and systems engineering standards for more than 20 years, through its Software and Systems Engineering Standards Committee (S2ESC). An overview of the universe of IEEE software engineering standards can be found in [12]. These standards cover the entire spectrum of the software engineering discipline. Currently, there are 27 active IEEE-CS working groups developing additional software engineering standards. A listing of these groups can be found at the website of the Society's

Standards Activities Board [13]. The topics of the working groups include test documentation, unit testing, and verification and validation.

Software engineering standards are also being developed on the international level. Here, the locus of development is a joint technical committee of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). ISO is a network of national standards bodies that coordinates national and private-sector standards in many different technology domains, and IEC is responsible for standards in electrical, electronic, and related technologies. In 1987, ISO and IEC established a joint technical committee (JTC1) to develop and manage standards in the rapidly emerging area of information technology. JTC1 has seventeen subcommittees dealing with standards in specific areas of information technology, ranging from coded character sets to biometrics. One of these subcommittees (SC7) is responsible for systems and software engineering standards. SC7 is in turn composed of many working groups that are responsible for developing standards on such topics as software process assessment and software systems documentation. Information about SC7 and its activities can be found at the SC7 website [14]. The IEEE Computer Society has Category A liaison status with ISO/IEC JTC1 SC7, and the S2ESC committee has sent many representatives to SC7 who are actively participating in SC7 working groups. Furthermore, S2ESC has been working to harmonize its software engineering standards with the international standards developed by SC7.

The rapidly growing collection of international standards for software engineering is consistent with the increasing professionalism of the software engineering discipline. The role played by the IEEE Computer Society in developing and promulgating these standards further reinforces this professionalism.

## ***2.5 Professional Societies and Bodies of Knowledge***

Professional practice relies on a commonly accepted body of knowledge. This is the case for traditional professions, such as medicine and law, and it is equally the case for emerging professions. For example the profession of project management rests on a body of knowledge that is codified in [15].

In 1999, two professional societies, the IEEE Computer Society and the Association for Computing Machinery (ACM), recognized that the evolution of software engineering as a profession called for the development of a body of knowledge, and initiated an effort to develop a Software Engineering Body of Knowledge [16] (Bourque 1999). This effort was crowned by the release of the SWEBOK Guide in 2004 [17].

However, it is important to note that the Association for Computing Machinery withdrew from the SWEBOK effort in 2000 [10]. This decision was based on ACM's conclusion that the body of knowledge was a step on the path to licensing software engineers, and that such licensing was premature.

Any body of knowledge must be subject to ongoing revision as a profession continues to evolve, and software engineering is no exception. The SWEBOK revision process is underway, and the next revision is scheduled for publication in 2009. The Software Engineering Body of Knowledge has received international recognition, and it has been published as a ISO/IEC technical report [18]. It serves as the technical foundation for the IEEE Computer Society's CSDP certification scheme.

### ***2.6 Professional Societies and Codes of Ethics***

As retired US Supreme Court Justice Sandra Day O'Connor said, "Professionalism requires adherence to the highest ethical standards of conduct ..." [3]. Consistent with this observation, professional disciplines have developed explicit codes of ethics, and such codes are commonly developed by professional societies. For example a rather generic IEEE code of ethics can be found at [IEEE code]. For software engineering, IEEE-CS and ACM have collaborated to produce an ethics code [19]. The Software Engineering Code of Ethics and Professional Practice [20] has been widely cited as providing an ethical foundation for software engineering practice [21]. Other codes of ethics produced by professional societies refer more generally to engineering or computing practice. An overview of codes of ethics for computer professionals and literature about such codes can be found in [22].

## **3. Conclusion**

The emergence of a professional discipline can be measured along several dimensions: governance, entrance control, preparation, body of knowledge, and ethical behavior. While the initial usage of the term "software engineering" may have been metaphorical, the ensuing 40 years have seen the creation of milestones along all of the dimensions mentioned above. It is therefore reasonable to conclude that software engineering has been moving toward internal and external recognition as a professional discipline, and that this progress can be expected to continue.

## **References**

1. Randell, B., Software engineering in 1968, In: Proceedings of the 4<sup>th</sup> International Conference on Software Engineering, pp. 1-10, IEEE Press, Piscataway, NJ, USA (1979).
2. Mahoney, M., The roots of software engineering, CWI Quarterly Vol. 3, pp. 325-334 (1990).



3. O'Connor, S. Court of Appeals of Maryland Professionalism Course for New Admittees to the Maryland Bar, In: Professionalism Above and Beyond Ethics vol. 15 (1992). Downloaded from <http://ilscpp.org/guidelines#ethics>, 17 Apr 2008.
4. Institute of Electrical and Electronic Engineers Code of Ethics (2006), downloaded from [www.ieee.org/portal/pages/iportals/aboutus/ethics/code.html](http://www.ieee.org/portal/pages/iportals/aboutus/ethics/code.html), 17 April 2008.
5. Shaw, M., Prospects for an engineering discipline of software, IEEE Software vol. 7, pp. 15-24 (1990).
6. Adams, T. Software engineering in Canada, the US, and the UK: inter-professional relations and the emergence of a new profession. Working Paper #9, Center for Workforce Aging in the New Economy, University of Western Ontario London, Ontario, Canada (2004).
7. IEEE Computer Society. Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Downloaded from [sites.computer.org/ccse/SE2004Volume.pdf](http://sites.computer.org/ccse/SE2004Volume.pdf), 17 Apr 2008.
8. Canadian Information Processing Society, Position Paper on Software Engineering (2000) downloaded from <http://www.cips.ca/it/position/softeng>, 18 Apr 2008.
9. Bagert, D., Licensing and certification of computer professionals. Advances in Computers, Vol. 60, pp. 1-34 (2004).
10. Association for Computing Machinery, A summary of the ACM position on software engineering as a licensed engineering profession (2000), downloaded from [www.cs.wm.edu/~coppit/csci690-spring2004/papers/selep\\_main.pdf](http://www.cs.wm.edu/~coppit/csci690-spring2004/papers/selep_main.pdf).
11. Seidman, S., An international perspective on professional software engineering credentials, in: Software Engineering: Effective Teaching and Learning Approaches and Practices, IGI Global, Hershey, PA, USA (2008).
12. Moore, J., The Road Map to Software Engineering: A Standards-Based Guide, John Wiley and Sons, Hoboken, NJ, USA (2006).
13. IEEE Computer Society Standards Activities Board website, at [standards.computer.org/sesc/s2esc\\_wkgroups/wkgndex.htm](http://standards.computer.org/sesc/s2esc_wkgroups/wkgndex.htm).
14. ISO/IEC JTC1 SC7 website, at [www.jtc1-sc7.org](http://www.jtc1-sc7.org).
15. Project Management Institute, A Guide to the Project Management Body of Knowledge, 3<sup>rd</sup> edition, Project Management Institute, Newtown Square, PA, USA (2004).
16. Bourque, P., Dupuis, R., Abran, A., Moore, J., Tripp, L., The guide to the software engineering body of knowledge, IEEE Software, vol. 16, pp. 35-44 (1999).
17. SWEBOK, The Guide to the Software Engineering Body of Knowledge, downloaded from [www.swebok.org](http://www.swebok.org), 18 April 2008.
18. International Standardization Organization, ISO/IEC TR 19759:2005: Software Engineering - Guide to the Software Engineering Body of Knowledge (SWEBOK), ISO, Geneva, Switzerland (2005).
19. IEEE Code of Ethics. Downloaded from [http://www.ieee.org/web/membership/ethics/code\\_ethics.html](http://www.ieee.org/web/membership/ethics/code_ethics.html), 18 Apr 2008.
20. Association for Computing Machinery, Software Engineering Code of Ethics and Professional Practice, version 5.2, downloaded from [www.acm.org/about/se-code](http://www.acm.org/about/se-code), 18 Apr 2008.
21. Anderson, R. The ACM code of ethics: history, process, and implications, in: C. Huff and T. Finholt, eds. Social Issues in Computing. McGraw Hill, New York, NY, USA, pp. 48-71 (1994).
22. Tavani, H., (1996), Professional ethics and codes of conduct for computer professionals (1996), downloaded from [cyberethics.cbi.msstate.edu/biblio/section03/chap01](http://cyberethics.cbi.msstate.edu/biblio/section03/chap01), 18 Apr 2008.