

In June 1998, the Texas Board of Professional Engineers established software engineering as a recognized engineering discipline and established licensing criteria specifically suited to software engineers. The author explains the legal issues involved and how they affect you.

What Do You Mean I Can't Call Myself a Software Engineer?

John R. Speed, PE, Texas Board of Professional Engineers

Your job title is *software engineer*. You've had the title for years. It's on your business card. However, somewhere along the line, you might have encountered someone who said that you couldn't call yourself a software engineer. They said it was against the law. Although you managed to ignore them at the time, you might have secretly wondered if they were right.

At the Texas Board of Professional Engineers, we regularly receive questions arising from such encounters. The callers come from every background—from high-tech to no-tech. Their education and experience levels vary from those without high school diplomas to PhDs, and from basic entry-level jobs to the highest technical ranks. Is it true that they can't call themselves engineers? The question seems straightforward, but the answer is complicated.

Two fundamental legal realities set the stage for the rest of this discussion. Only licensed professional engineers

- ◆ have the legal authority to use an engineering title without restriction, and
- ◆ can offer engineering services.

CERTIFICATION IN THE UNITED KINGDOM

Trevor Burridge

Engineering accreditation in the UK is governed by the Engineering Council, which acts as an umbrella organization for all the individual engineering professional bodies, including the Institute of Electrical Engineers, the British Computer Society, and the Institute of Mechanical Engineers. Each institute has a number of professional or corporate membership grades (member or fellow). For those that cannot qualify for professional membership, there are other grades, such as associate member or companion.

To qualify for professional membership, applicants must demonstrate a sufficient combination of education (usually good grades in an accredited bachelor's program), training (usually two years of approved training), and relevant experience in a position of responsibility. Substantial experience can often be offered in lieu of formal training and, in some circumstances, even education. Applicants must be sponsored and proposed by several existing professional members and must undergo a technical interview by a panel of experts.

Corporate members of each professional engineering organization are eligible to be entered on the Engineering Council Register and can only then use the designator CEng (Chartered Engineer) after their names. Engineers who fail to meet accepted professional standards or fail to abide by the code of conduct and ethics can face disciplinary action, which might result in them being taken off the register. This form of certification is the same for all branches of engineering in the UK.

The IEE and BCS are both full members of the Engineering

Council. For almost a decade, both these bodies have been accepting and awarding corporate membership and Engineering Council registration to software engineers. For more information about IEE and BCS membership and entry requirements, see <http://www.iee.org> and <http://www.bcs.org.uk>. The Engineering Council Web site is <http://www.engc.org.uk>.

A chartered software engineer therefore has similar entry requirements and standing as a chartered civil or mechanical engineer. However, from my experience, only a small minority in the software development field in the UK bother to apply for any form of membership, let alone submit to the entry procedures for professional membership. I believe the main reason for the low uptake is that employers do not see it as mandatory or even especially desirable. Job advertisements focus almost exclusively on the technical skills with the three-year half-life in C++, Java, NT, UML, Orbix, Oracle, and so on. (See <http://www.jobserve.com> for typical UK software jobs. Search for "software engineer" and see what you come up with.) This is even true with more "enlightened" employers who are genuinely attempting process improvement initiatives.

I suspect part of the problem is the chicken-and-egg situation. Employers think that there are not many chartered software engineers, so they see no point mentioning this as desirable in job advertisements. Applicants see that chartered status isn't important to get a job, and it gives no extra benefits or pay, so why bother with the hassle of becoming chartered?

The UK has had this version of full engineering certification for software engineers for a decade now, but it is still not a factor that most practicing software developers consider important.

Trevor Burridge is a freelance consultant. Contact him at trevor@parallel-consultants.com.

Because the term software engineering is so common, the Texas Board of Professional Engineers felt that existing restrictions on the use of engineering could grow into an undue business hardship on software engineering practitioners. In June 1998, the Board established software engineering as a recognized engineering discipline and established licensing criteria specifically suited to software engineers.¹ The intent was to streamline the licensing process for a growing new field of engineering.

To get a better understanding of the situation, let's take a crash course on licensing, title use, and engineering practice.

WHAT DO ENGINEERING LICENSING LAWS REGULATE?

In most US states, laws prohibit the unlicensed "practice of engineering." Texas law is typical, so we'll use it as an example. (See the two sidebars for in-

formation on licensing and certification information in British Columbia and Ontario and in the UK.)

Regulation of engineering practice starts with a legal definition. The definition of the "practice of engineering" generally looks something like this:

The "practice of engineering" means any service or creative work, the adequate performance of which requires engineering education in the application of special knowledge of the mathematical, physical, or engineering sciences to such service or work.²

Despite its circular nature, the definition has several important features.

- ◆ Engineering practice is defined in terms of an engineering education.
- ◆ Engineering practice can be recognized by its application of mathematical, physical, or engineering sciences to a problem.
- ◆ Any problem that cannot be adequately

BRITISH COLUMBIA AND ONTARIO LICENSING

Keri Schreiner

In recognition of the often-unconventional path to software development expertise, the Professional Engineers of Ontario organization has announced new criteria for licensing software practitioners as professional engineers.

Traditionally, an applicant for licensure must have experience and education in the same area. However, the dearth of accredited software engineering programs makes it difficult for software practitioners to meet the criterion. To address this issue, PEO has until now assessed applicants for software engineering licensure on a case-by-case basis.

PEO President Patrick Quinn said that their new criteria defines the core knowledge required for PEng licensing and provides a way to consistently assess practitioners' qualifications. "This is an important change and a first step in introducing professional regulation to the software industry," Quinn said, adding that the Y2K problem has highlighted the need for such regulation and professional accountability.

Provided they meet general criteria, candidates without a degree from an accredited engineering program or equivalent can now be licensed by PEO if they pass written exams or prove relevant work experience beyond the four years required under general licensing criteria. Other general criteria include knowledge of control theory, mathematical foundations, digital systems and computer architecture, and software design and programming

fundamentals. Candidates must also show knowledge in three of seven specialization areas and successfully complete the Professional Practice Examination on engineering law and ethics.

PEO's announcement follows an earlier one by the Association of Professional Engineers and Geoscientists of British Columbia, which said in June that it would begin licensing software engineers. The APEGBC's qualifications are a bachelor's degree or higher in electrical or computer engineering, engineering science, physics, or the computer-software field and a minimum of four years of software engineering experience.

Like PEO, the APEGBC cited two primary factors in its decision to extend licensure to software practitioners: their important contributions to the engineering field and the need for professional regulation in the software industry. Stuart Bourhill, chair of the APEGBC's Computer and Software Engineering Task Force, said they have been working closely on licensure issues with the Canadian Council of Professional Engineers, the IEEE, and several Canadian universities. "We believe that only through close cooperation with these other groups and professional societies will we be able to adequately address the needs of the software engineering professional," he said.

The APEGBC also announced the formation of a new division for computer and software engineering that will provide a forum for practitioners to discuss professional guidelines, development strategies, and education opportunities. More information on both announcements is available at the PEO Web site (<http://www.peo.on.ca>) and the APEGBC site (<http://www.apegbc.ca>).

Keri Schreiner is a contributing editor for *IEEE Software*. Contact her at keri@grooveline.com.

solved without applying an engineering education to it, by definition, an engineering problem. US states create licensing boards to establish a system to license qualified engineers and regulate their practice.³ By law, those rules must concentrate on the protection of public health, safety, and welfare.

Some software engineering activities clearly fall under the legal definition of the engineering practice. For example, the design, testing, and implementation of embedded and real-time systems require a detailed understanding of the engineered electrical or mechanical components. Similar activities performed on software systems for mechanical devices, electrical devices, and power systems are clearly engineering services.

Other activities are harder to classify as engineering or nonengineering, primarily because of

the vague nature of the phrase "public health, safety, and welfare." Although almost everyone understands the need to protect public health and safety, many gloss over the term *welfare*. After all, it's easy to justify regulations that are designed to keep water clean or prevent explosions. But what does welfare mean in this context? It describes a state of general well-being or prosperity. I suggest substituting words and phrases such as money, resources, or business interests in place of welfare. Instead of protecting public health, safety, and welfare, the regulations are protecting the public health, safety, and business interests of citizens. Because of the many ways in which software affects welfare and the term's encompassing meaning, you can see why most enforcement cases the Texas Board investigates involve engineering's impact on welfare.

RESTRICTIONS ON ENGINEERING TITLES

Licensing laws are designed to maintain a level of integrity in the engineering profession, restricting use of the term *engineer* to people who have demonstrated a minimum competency.²

Engineer title restrictions are most strict in situations where someone is attempting to offer consumers consulting or contract services. They are also strict in situations where engineering is practiced in the public sector. Licensing boards have designed title restrictions in these circumstances to protect consumers against fraudulent claims. The laws operate on the fundamental premise that a US state government should provide some level of consumer protection against fraud, and they are aimed at those who falsely claim engineering expertise.

If you are reading this article, you are probably most concerned with a few specific titles such as software engineer, systems engineer, or network engineer. Licensing boards are concerned with *any* use of the title engineer. As I mentioned earlier, US state licensing boards allow unlimited use of engineering titles only if you are licensed as a professional engineer in that state.³

A person can also break laws related to the use of

Regardless of what you call it, offering engineering work without a license is illegal.

the term engineering in two ways. One way is to offer or perform engineering services without obtaining the required qualifications. The other is to incorrectly label an activity engineering. In Texas, we see the law broken in both of these ways.

UNQUALIFIED SOFTWARE ENGINEERING

It is illegal to practice software engineering in Texas without a license (or an appropriate exemption, which I'll discuss later). As I mentioned earlier, development of software for engineered systems—including embedded systems, real-time systems, mechanical devices, electrical devices, and power systems—is software engineering. A person who performs software development work in these areas without a professional engineering license or ex-

emption is breaking the law. Many practitioners try to skirt the law by calling their work “technical services” or “technical consulting” instead of “engineering.” Regardless of what they call it, offering engineering work without a license is illegal.

CALLING SOFTWARE DEVELOPMENT SOFTWARE ENGINEERING

Many kinds of software development affect the public welfare but are not considered software engineering under Texas law. Work on financial systems or business systems might affect the public welfare, but it doesn't require an engineering education, doesn't require the application of physical sciences, and can be performed without an engineering education. (See this issue's Guest Editors' Introduction for ways in which this educational situation is changing.) Unless these activities also include the design of hardware interfaces, power systems, or other engineered components, they do not appear to meet the legal definition of engineering practice.

When someone reports illegal software engineering practice, the Board's first task is to determine whether the practice is really engineering. Quite often, our evaluation determines that the activity in question is not. It might be more accurately described as “code writing,” “product support,” or other tasks that the board doesn't regulate. If the practitioner is calling these activities “code writing” or “product support,” there are no legal problems. However, if the practitioner is calling the activity software engineering, he or she is violating the law. Such use of an incorrect term leads consumers to believe that they are receiving services in full conformance with the legal protections provided by engineering law. Therefore, anyone using such terms illegally should reasonably expect legal problems.

EXEMPTIONS

Legal use of an engineer title depends on two variables: how you use the title and the employment conditions under which you practice engineering.²

Almost every US state law has some allowance for the legal use of engineering titles by nonprofessional engineers,³ or *exemptions*. The *industrial exemption* is the most commonly claimed. It is avail-

able to a well-defined class of engineers, which happens to be the largest class of engineers in the nation. Engineers can claim an industrial exemption if they meet the following conditions:

- ◆ They practice engineering only for their full-time employers.
- ◆ Their practice is limited to work on their employer's facilities or on products that their employer manufactures.
- ◆ They do not use an engineering title outside their company.
- ◆ They do not claim that they are qualified to offer engineering services to another party.

Let's imagine that you meet the requirements for an industrial exemption. How do you avoid title-use problems? Most of the illegal title-use cases in Texas stem from situations where practitioners misunderstand the industrial exemption. The best way to avoid problems is to practice "title abstinence." Never refer to yourself as an engineer outside your company. Never moonlight engineering work for another company or project. Never offer engineering work on a contractual basis.

Most US states have other title exemptions, which are available to individuals such as full-time federal employees, railroad engineers, and engineers working for utility companies. Graduates of recognized engineering programs are often allowed to use a term such as *graduate engineers*.² All states have programs to recognize those who pass the first national examination, the Fundamentals of Engineering exam; they can register in their state to use terms such as *engineering intern* or *engineer-in-training*.³

As with titles, there are industrial, federal, and other exemptions for engineering practice, and the circumstances are identical. If you perform your engineering work only for your full-time employer's property or on their manufactured products, and you do not violate title restrictions, you are eligible for an exemption.

HOW SOFTWARE ENGINEERS ARE LICENSED IN TEXAS

For almost four decades, the Texas Board has licensed software engineers under other discipline headings, such as electrical, industrial, or control-system engineering. The reason for licensing under these other disciplines is that software en-

gineering activities were generally performed in support of one of these more "traditional" engineering disciplines.¹

Once a license has been issued, the discipline designation is almost totally irrelevant. In most states, individuals are licensed as *professional engineers*, not as software engineers, chemical engineers, electrical engineers, and so on.⁴ In the same way that statutes require attorneys or medical doctors to practice only in their areas of competence, professional engineers are required to limit their practice to their areas of expertise. Obviously, as engineers develop new skills, their area of expertise might change.

At a meeting in June 1998, the Texas Board enacted rules that recognized software engineering as a distinct engineering discipline,⁴ which lets individuals with acceptable educational credentials and software engineering experience apply for a license.

Because no nationally recognized software engineering exam exists, only highly experienced software engineers are eligible.¹ The licensing scenario outlined below is the only route available to a Texas software engineering license until software engineering exams are developed:

- ◆ Possession of an engineering, a computer science, or other high-level math or science degree evaluated by the Board as adequate.
- ◆ At least 16 years of creditable experience performing engineering work (only 12 years are required for individuals holding a degree approved by the Engineering Accreditation Commission of the Accreditation Board for Engineering and Technology).
- ◆ References from at least nine people, five of whom must be licensed engineers.
- ◆ Required educational and other credentials.

Anticipating need, the Texas Board has also initiated the development of exams for licensing software engineering practitioners with less experience. It intends to seek national use of any exams through the National Council of Examiners for Engineering and Surveying. Once the exam is available, less experienced individuals can apply for a Texas PE license by passing it. Eight years of the experience requirements may be subtracted for those who pass an NCEES exam.

The best way to avoid problems is to practice "title abstinence."

Should you pursue a PE license? Everyone has personal career goals. Are you doing work that meets the legal definition of engineering practice? Do you plan to work in that arena during your career? If so, you should contact your state licensing board for details about how to get started.

Licensing is one of the logical developments of any maturing profession. Software engineering is far more mature than many practitioners realize. As you ask yourself whether you should pursue a professional engineering license, look around your office. In 10 years, what will distinguish you from the people around you? What will set you apart as a professional? For most of this century, the answer has been the professional engineering license.

Licensing boards around the nation will address the need to license software engineers as exams are developed. This process will take several years, but we are moving ever closer. Decision time is approaching. How will you respond? ♦

About the Author

John R. Speed is a consulting engineer specializing in design and program management for public infrastructure and a frequent lecturer on engineering ethics and professional practice. He is the former executive director of the Texas Board of Professional Engineers, where he managed the state's regulation of the engineering profession. He holds a BS in civil engineering from Texas A&M University, an MA in political science with an emphasis in public management from Midwestern State University, and is a graduate of the Texas State Governor's Executive Development Program. He is a member of the National Society of Professional Engineers and the American Society of Civil Engineers.

REFERENCES

1. J.R. Speed, "Software Engineering: An Analysis of the Proposed Actions by the Texas Board of Professional Engineers," Texas Board of Professional Engineers, Austin, Texas, 1998.
2. The Texas Engineering Practice Act, Article 3271a, *Texas Civil Statutes*, 1937 (Revised in 1999); <http://www.main.org/peboard/law.pdf> (current Oct. 1999).
3. "Summary of Engineering Licensure Laws (Part IIB, United States)," *Engineering Licensure Laws*, National Society of Professional Engineers, Alexandria, Va., 1999.
4. Practice and Procedure, Bylaws and Definitions, Title 22, *Texas Annotated Code*, Chapter 131. <http://www.main.org/peboard/law.pdf> (current Oct. 1999).

your Vision
our Future

Computer Scientists & Engineers

Southwest Research Institute has exciting opportunities available in Modeling and Simulation, Software Engineering, and Software Design. All positions require a BS or MS in Computer Science, Computer Engineering, Electrical Engineering, or equivalent; programming in C, C++, and Java on PC- or UNIX-based workstations. Responsibilities include software requirements generation, design, prototype development, testing, implementation, integration, maintenance, and support; interaction with Institute staff and clients; and travel to client sites.

Modeling and Simulation

Analyze, design, develop, and implement modeling and simulation algorithms and software for training systems. Requires experience using Ada; experience with distributed simulation (live, virtual, and constructive); knowledge of computer systems, software architectures, hardware-software interfacing techniques; and the ability to work with MS Word, Excel, and Project.

Software Engineering & Design

Design and develop software for military and industrial applications including real-time data acquisition, graphic user interfaces, system level control processing and embedded algorithm implementation, and network interfaces (Internet, LAN/WAN, etc).

Please submit your résumé to: Southwest Research Institute, Human Resources Dept. 00103, P.O. Drawer 28510, San Antonio, TX 78228-0510
E-mail: personnel@swri.org • Fax: (210) 522-3990

www.swri.org/jis6/

An Equal Opportunity/Affirmative Action Employer • A Public and Community Services Employer

- Competitive salaries
- Medical, dental, & vision care
- Retirement & benefits
- Life & long-term disability income
- Educational advancement
- Professional development
- Awards for individual research
- Appointment & relocation
- Medical clinic
- Employee assistance
- Generous vacation
- Ten paid holidays
- Credit union
- Dining services
- Travel service
- Fitness facilities



Southwest Research Institute