DRAFT Proceedings

Should Software Engineers be Licensed Engineers?

D. Emery, B. Thomas, organizers

In spite of the millions of software professionals worldwide and the ubiquitous presence of software in our society, software engineering has not yet reached the status of a legitimate engineering discipline and a recognized profession.

Since 1993, the IEEE Computer Society and the ACM have been actively promoting software engineering as a profession and a legitimate engineering discipline, notably through their Software Engineering Coordinating Committee (SWECC).

ACM has recently adopted positions in opposition to the licensing of software engineers as Professional Engineers, and has withdrawn from the Software Engineering Body of Knowledge (SWEBOK) project. (See http://www.acm.org/serving/se_policy/). These are important issues for the software community. The ACM position was introduced during the Wednesday Morning Plenary session, and was further discussed during this Workshop.

A few quotes from the ACM positions help set the context. From http://www.acm.org/serving/se_policy/report.html:

> "[F]ramework of a licensed professional engineer, … does not match the professional industrial practice of software engineering. Such licensing practices would give false assurances of competence even if the body of knowledge were mature; and would preclude many of the most qualified software engineers from being licensed."

From http://www.acm.org/serving/se_policy/bok_assessment.pdf:

> "[C]urrent software engineering body of knowledge efforts, including SWEBOK, are at best unlikely to achieve … appropriate assurances of software quality…"
>
> "ACM's continued participation … will not further - and, indeed, may distract from, efforts to improve software quality…"
>
> "[W]ithdraw ACM from further involvement with SWEBOK…"

The question posed for this SIGAda Workshop is "Are ACM and SIGAda members well served by these ACM Positions?' A key goal for the workshop was for the participants to understand the Professional Engineering licensing process. To that end, representatives of the Society of Professional Engineers (http://www.nspe.org) were invited to participate in the workshop.

The workshop was attended by <paste list>.

The workshop opened with a talk by Patrick Natale, Executive Director <check…> of NSPE. Patrick provided background on the licensing process. Engineering is defined as "the application of scientific knowledge to practical problems." NSPE has proposed that engineering include "developing computer codes for engineering applications." This prompted a discussion of what constituted "engineering applications." CAD systems, as systems that support the work of "classic" engineering disciplines (structural, mechanical, electrical, etc.) could be considered as 'engineering applications'. Paul Stachour points out that other kinds of software systems could also be considered as 'engineering applications', particularly if they directly affect the health or safety of the public. One example that was mentioned was Air Traffic Control systems. Within the workshop, there was a general consensus that a definition of "engineering applications" should be interpreted broadly, to include both software in support of traditional engineering functions and software that provides critical functions for the public welfare.

Note that there are many other emerging engineering domains, such as biomedical engineering or genetic engineering, that could also be considered for licensing under a Professional Engineering licensing program.

There are 3 steps to licensing for professional engineers:

1. Education (accredited degree program)
2. Experience
3. Examination

Each state has its own licensing laws and regulations. Some states identify the engineer's specific area of competence; others simply license "Professional Engineers" and expect that the individual engineer will work only where she/he is competent. Some, but not all, states have continuing education requirements as part of licensing. Each state has a board that governs the licensing and conduct of Professional Engineers licensed in that state. This board is populated by engineering professionals using procedures specified by the state.

The role of a 'body of knowledge' is generally important for establishing the knowledge required for licensing. This is so that the public and other engineers can gain trust and understanding that a licensed engineer knows the principles of engineering in his/her domain. But a formal 'body of knowledge' document is not a requirement for licensing.

It is not a requirement that every individual doing engineering be licensed. In particular, corporations (and government agencies) can have a group of people doing engineering, as long as this group is supervised by a licensed engineer. But ultimately, there is a licensed engineer that would have to assume overall liability for the project. Professional Engineers have a seal that they affix to drawings, that certify that the Professional Engineer is personally responsible for the contents of the work. The standard of responsibility is not absolute. If a building falls down, the Professional Engineer that sealed the drawings is not automatically liable. If the building was engineered using reasonable techniques and common practices, the engineer is not automatically assumed to be liable. To be liable, the engineer must have made "aggregious errors", that the engineer should have known about, as a consequence of his/her knowledge and

experience. An engineer who makes such errors is subject to loss of licensing by action of the state licensing board, as well as other legal actions. Thus a common understanding of engineering practices is required for the Professional Engineering licensing process.

Robert Lief from AdaMed spoke next. <get slides> He talked about his experiences in the biomedical technology industry. He pointed out the need for good education and a standard curriculum in software engineering. Specific programming skills (C, C++, Java, etc) should not be a requirement for licensing. If very strict criteria were applied for software engineers, maybe 90% of current professionals would lose their jobs, and we can't afford to close down Microsoft and the rest of the software industry.

Lief talked about the importance of safety in embedded systems . He cited the FDA's hazard analysis program, pointing out that the final certificate for a medical system is signed by an MD, and not a software or systems person. From this he points out the difference between domain knowledge and software engineering knowledge. The domain expert has no formal relationship with the customer. If the domain expert gives the software engineer incorrect domain information, the software engineer is the one that gets blamed by the customer. Thus you need two experts to review work, one from the domain side and the other from the software/system/implementation side.

Paul Stachour pointed out the lack of general knowledge about 'good tools' in software. He cited an example where 98% of the students in a professional level course (BS in CS or engineering with 3-5 years work experience) used the wrong tool for analyzing concurrency. He also mentioned that the reason we can't all use a single common programming language is that the language must be suitable for the problem domain.

Jim Moore from MITRE provided information on the SWEBOK project. This was originally a project of a joint ACM/IEEE effort on professionalism. That effort produced the Code of Ethics adopted by both ACM and IEEE Computer Society. The intent for the SWEBOK is to capture "generally accepted engineering principles" as applied to software. At first Moore thought this would be used to prepare software engineering licensing or certification exams, but this is not the case. (More on this topic later.) Instead, the SWEBOK will probably be used most for education and training, and in particular to establish standards for academic program accreditation. It could also be used as a basis for continuing education, if that becomes a formal requirement. Thus the effect on licensing by the SWEBOK would be indirect. As SWEBOK influences curricula, this will establish what software engineers know, and that is what will influence what goes into the software engineering examination.

Joyce Currie Little pointed out that the ACM position was generally to "lay low" on these issues. ACM "insiders" tend to be prejudiced their academic (versus practitioner) orientation. Thus they have a vested interest in maintaining the status quo with respect to Computer Science (as an academic discipline) versus Software Engineering as a different academic and professional discipline. Moore points out that there is a great fear of licensing, that it will prevent current people from continuing to do what they currently do. Using the point that Patrick Natale established, it is likely that large development

organizations can have one or two licensed engineers supervising the work of a much larger group of developers.

Moore then asked Natale to talk to how licensing exam questions are generated. Natale said that there is a National Council for Examining Licensed Engineers and Surveyors. It is made up of representatives from the (75 different) state boards that cover engineering and surveying licensing. Test questions are formulated by this group, composed of experienced Licensed Engineers. There is a 5-year process for introducing questions into the licensing examination. An issue for any new field (such as software engineering) is establishing the experts who develop the questions. For a software engineering exam, the initial experts would be Licensed Engineers (from existing specialties) who also have substantial experience in software engineering. Eventually, the process will bootstrap as more licensed software engineers gain the experience and standing to participate in the National Council efforts. Natale also pointed out that the NSPE trains the examiners who participate in school accreditation under the ABET <get acronym> program.

Hal Hart spoke next, as a representative from 'industry'. Hal says that he agrees with the ACM assessment of the SWEBOK as immature, and that licensing software engineers is premature. But Hal sees no reason for ACM to pull out of these processes. The current SWEBOK is a good start, but needs more iterations. In particular, Hal cited the large amount of material that is included "by reference". Moore responds that the SWEBOK group is trying to get copyright to place such material "by value" into the SWEBOK document. Moore points out that the current document is the result of 3 years work and the next version is a "stoneman" version for trial use.

Hart says that licensing is premature without a clear consensus on the body of knowledge. There is no basis for 'good engineering practice' or for developing software engineering test questions without a clear body of knowledge. But Hal strongly disagrees with ACM's withdrawing from these processes. He believes that the IEEE's basic process is sound, and would be better with ACM participation. ACM should continue to improving both the process and the content. And, Hal insists that ACM members should participate, even without ACM Council participation/sanction. Hal says that licensing is "inevitable", and it would be better for the professional community, including ACM, to define the criteria, rather than to let this be done by politicians or other individuals not familiar with software engineering. But we need to come up with a clear statement of "scope" for licensing. Should this apply to engineers doing "safety-critical" systems? This is an important consideration for where we go with licensing.

Dave Emery asks: If we have licensing, then what do licensed engineers do? Specifically, if the reason for licensing is to assume responsibility/accountability for software, how do we establish this? Will we have 'licensed software engineers' seal listings, the same way that civil engineers seal blueprints? Currie Colkit points out that in the UK there is a procedure for a 'responsible person' to assume such responsibilities for software.

Frances Anderson points out that we have to get both management buy-in and team buy-in. Karen Moran (PE Civil Engineer, from NSPE) points out that she must have confidence that her team (as many as 50 engineers/designers) has done the right thing before sealing the drawings. Jim Moore points out that we are trying to establish a chain of responsibility, and this would be A Good Thing for software. Moran says that the standard of care is not perfection, but what would another reasonable engineer do in the same situation.

Ron Oliver asserts that what we do in software is very different in many ways from civil engineering. Emery responds that the primary difference is that we are constrained by mathematics, rather than physics/mechanics of materials. Otherwise what we do is very similar to civil engineering. Bridges aren't supposed to fall down; programs aren't supposed to fail. Stachour mentions the standard of care for testing, which is "well known". (Emery: There are analogies with testing in civil and electrical engineering.)

At the end, Hal Hart proposes to capture the consensus of the workshop with 3 resolutions. These were approved, with the votes as indicated (for/against/abstain)

1. General principle is to see ACM strengthen support for professionalization activities in software engineering. 13/0/1

2. ACM council should reconsider decision on withdrawal from collaboration with IEEE on SWEBOC/software engineering coordinating committee. 12/0/2

3. ACM serves its members better by being involved in activities associated with the potential licensing of software engineers. 13/0/2

Hal suggested that SIGAda could forward these positions to ACM Council.