Subject: [Fwd: RE: **SE Licensing Panel conf call Monday 5PM EST]**
Date: Wed., 14 Apr 1999 14:09:23 -0500
From: "Dennis J. Frailey" **<d-fraileyCalraytheon.com>**
Organization: Raytheon Systems Company
To: Mark Ardis <maa@bell-labs.com>,
Dennis J Frailey <d-frailey@raytheon.com>,
Linda Northrop <lmn@sei.cmu.edu>,
Doris Carver <d.carver(alcomputer.org>, Karl Reed <k.reed@computer.org>,
Leonard Tripp <Leonard.Tripp@PSS.Boeing.com>

-------- Original Message --------
Subject: RE: SE Licensing Panel conf call Monday 5PM EST
Date: Sun, 4 Apr 1999 12:28:38 -0500
From: "Marry, William" <WMarcy@coe.ttu.edu>
To: "'edd'" <dandm@texasonline. net> "David Rentschler,PE"<David.Rentschler@compaq.com>,"Marry, William" <WMarcy@coe.ttu.edu>,"Otto M. Friedrich, Jr." <Otto Friedrich@ mail.utexas.edu>,Dennis Fraitey<D-FraileyCalraytheon.com>,"Gerald Burnham,, P.E. if <burnham(alutdallas.edu>,Donald Bagert <bagert@ttacs.ttu.edu>

It's hard refute Dave Parnas' arguments. We need to do what is right. We must not let the existing "Craft-Guild" approaches to the practice of software engineering take precedence over the public interest. The practice of medicine is far from perfect. I don't think the public would accept the impossibility of perfect medical practice as a reason to "de-regulate" the practice of medicine.

-----Original Message ----
From: edd [mailto:dandm(d) texasonline.net]
Sent: Saturday, April 03, 1999 8:00 AM
To: David Rentschler,PE; William Marry, P.E.; Otto M. Friedrich., Jr.; Dennis Frailey; Gerald Burnham., P.E.; Donald Bagert
Subject: Fw: SE Licensing Panel conf call Monday 5PM EST

Gentlemen - There was a conferennce call last week between a task force ACM set up to study software engineering licensing and the Texas Board (Speed, May and Dorchester). I thought you would be interested in an E-mail from

Dave Parnas. He was scheduled to be included in the conference call in fact I think he is on the ACM task force - but was not available at the time so they recorded the conversations to forward to him. He has made some comments based on the tape and I thought you would be interested in them. I would be interested in any comments you might have. Thanks for all of your help.

Dave Dorchester, P.E.

> From: Dave Parnas <parnas@9usuntcas.mcmasterca> > To: Farber@ds. upenn. edu; Gtay@Microsoft. com; bhuff@nae. edu; boalen@hq.acm.org; boehm@sunset.usc.edu; brooks@cs-unc.edu; browne@cs-utexas-edu; butler@rice.edu; dandm@texasonline.net; dnagel@att.com; franalle@US.IBM.COM; graham@cs.berkeley.edu; john .speed@ mail.capnet.state.tx.us; ken@cs.rice.edu; leveson@mit.edu; neumann@sri.com; parnas@mcmail.CIS.McMaster.CA; pbhawthorn@mindspring.com; simons@ACM.ORG; waiter. may @ mail .capnet.state.tx.us; white@ACM.ORG; wwulf@nae.edu *> Subject: Re: SE Licensing Pane/ conf call Monday 5PM EST >Date: Friday, April 02, 1999 11:57 AM*

Last night I had the opportunity to listen to the tape of the
> conversation that some of you had with the Texas Board of Engineers
> and I made a few notes I am afraid that I am not great at
> "voice recognition"so I am not sure who said what. I made
> a few notes

> 1. Perfection is the enemy of the good.

> A constant theme from people with *non-Texas accents was that it > is beyond the* state of the art today to make error-free software *> or software with the "quality that one would expect of a professional engineer': > I consider this a "red herring': It may come as a surprise to > some of you, but we don't seem able to design perfect cars* > (witness *the recalls), perfect radios, perfect buildings (come > and see my office) or perfect anything else. Despite our > inability to build perfect products, we continue to build them*

and should be trying to improve the situation. I have to believe that people who have met a well-thought-out set of educational requirements and have appropriate guided experience will do a better job than someone who could not find a job as a historian and so went into software without any education on that topic.

2. A frequently repeated statement was something to the effect that 'The Body of Know/edge isn't there': With apologies to those who said that, I think it is nonsense. There is a lot of know/edge (science, computer science, mathematics, certain engineering areas) that should be known to a software developer and most of it is not known. Sure, there is a/ways a need for more research, (which is why I remain an active researcher) but today the greatest need is for education and technology transfer. The bad systems that 1 see, and lots of papers, show a lack of know/edge of ideas that were well understood 10 years ago.

In fact, when we developed our 4-year SE programme here at McMaster we had to leave a lot out. We collected what we wanted to teach and thought relevant and then the hard work began as why tried to figure out what part of that we could fit into a tightly packed programme. Moreover, none of it is transient know/edge. It is all material that we believe would have been useful if we had known it 20 years ago and will be useful decades from now. The only p/ace where we have producttechnology dependent material is in the laboratory portion of some courses where they have to use some products. My analogy, from EE, is a circuit theory course. Most of what I /earned insuch courses 40 years ago i5still useful today. In the labs, I /earned how to struggle with primitive "scopes': That is useless today. We have the same distinction in our programme.

The body of know/edge is growing and, to a lesser extent, changing just as it is in EE or Civil Engineering, but that is certainly no reason not to institute licensing. As the body of know/edge grows so do our expectations but one of the expectations of profesSional engineers is that they keep up with the changes.

I also disagree with the often repeated statement that engineers do not understand the rapid rate of growth in software technology. Often that is all that they understand. What they do not understand is the fundamentals.

I observe that what is worth knowing in EE has changed as much in my working lifetime as what is worth knowing insoftware. When 1 started my Ph. D., there were 'advisors" who told me to avoid semiconductor research because it would never be practical,

never be able to handle audio /eve/ power, etc. Think of the change since that time. When 1 worked as a radio engineer, we were told that trees could be avoided in antenna design. Today, that is false. Change is rapid in all areas of Engineering and licenses are still valuable.

.3. Different types of software

Another remark was that we can t license because there are too many different types of software. Again, this would pule out licensor in other areas of engineering. There are many different types of buildings, bridges, circuits, etc. A power p/ant designer and a circuit designer have very different constraints. However, the fundamental laws (Maxwell laws, Kirchoffs laws) are the same for both. The mathematics is often the same for both. The methods are often very dose. This is true for software. Much of what we can teach applies to many types of software.

4. Get rid of the worst, identify the best

We all know many incompetents who describe themselves as software engineers, do not bother to read or learn, and do a terrible job. Licensur would not get rid of all of them, but it might get rid of some of the worst or force them to learn something. It would also identify some of the best.

5, Beneficial effects even when license not required.

Because it is their job, the Texas Board people focus on the situations in which a license is required or people practice without a license. However, in Engineering the licensur system has beneficial effects even though most graduates do not get a license. It has encouraged educators to think about defining the core body of know/edge in each field. and to make sure that people learn it. People working for a license learn important things even if they never use the license. Some employers want either PE or PE-qualified people because they know what they know and fee/ they can trust them. If l were hiring an engineer today and had a choice between one with a license and one without, and other things were about the same, I would take the licensed one.

6. Educational programmes

Someone mentioned that the web-site contains pointers to SE programmes. The issues can be understood in more depth

if you look at those. In particular, I hope you will look at the McMaster programme. It is one of the first to be design for accreditation in Canada. It is the only one that is not a mixture of existing CS and Computer Engineering courses t contains all new courses aimed at people who will be profess engineers specialising in software. You should also look at the programme at the University of Ottawa. This program.
also appears to meet licensing requirements but does it by using a lot of older CS and CE courses

In our education programme you will find some thing-5 that a. not universally required. Even in my EE programme there were things that only a fraction of us would use. However, in our programme I think you will find little that will get out of date.

7 The licenser requirement is there now.

The call makes it clear that the license requirement is there now. It may not a/ways be enforced and it is clear that the criteria may not be right, but ACM cannot decide whether or not there should be licenser. What ACM can do is help to improve the system by working on the identification of the core body of know/edge. I have looked at the IEEE committee and believe that they need more ACM input. In particular, th need input from scientists to make sure that the "core body" includes fundamentals rather than fads and folk/ore.

Prof. David L Lorge Parnas, P. Eng. NSERC/Bell Industrial Research Chair in Software Engineering Director of the Software Engineering Programme DEPARTMENT OF COMPUTING AND SOFTWARE Faculty of Engineering McMaster University, halmilton Ontario Canada L8S 4L7

Telephone: 905 525 9140 Ext 27353 Telefax.,
905 525 6246 email., parnas@qusunt, CAS.
McMaster. CA