

Dave Parnas
McMaster University
parnas@mcmaster.ca
<http://www.crl.mcmaster.ca/People/Faculty/indparnas.html>

The rationale for licensing "software engineers" is exactly the same as the rationale for licensing other engineers. In both cases, poor design can often endanger the safety and well being of the public. Consequently, it is vital to make sure that those who design these products are fully qualified by their education and experience to do that kind of work. Moreover, it is important to be sure that they are fully aware of their professional and legal responsibilities. Finally, it is important to make sure that some authority can investigate and, where appropriate punish those who are negligent and unethical.

The mechanisms needed for licensing need be no different for software engineers than they are for other engineers. The licensing authority must be created by legislation and have the authority to enforce the legislation. In particular, they must have the authority to remove a license from licensed practitioners who do not follow established rules of good practice, and they must have the ability to stop those who practice without a license in situations where a license is required.

The most efficient way to establish meaningful licensing for software engineers is to recognise it as a new speciality within engineering. Engineering has many specialities. There is a core of knowledge expected of all engineers; this is supplemented by specific knowledge needed in each speciality. The most difficult task facing us in the licensure of software engineers is the identification of the special knowledge that should be required of those who practice in this field.

The other task required is the characterisation of those types of software development that must be under the control of a licensed Engineer specialising in software. In those cases it is not necessary that all employees be licensed software engineers, but the final product should bear the seal of a licensed engineer who takes responsibility for the product's being fit for use.

ACM and IEEE can be of help to the Professional Engineering Societies in these two tasks.

A committee can propose a description of the essential knowledge for a software engineer. This description can NOT be determined simply by asking today's programmers what they know. Most of them do not have the appropriate training, skills, or knowledge. The committee must ask experts what program designers SHOULD know in order to make sure that their products are fit for use.

Another committee can examine the many applications of software and determine which of these are so critical that they should bear the seal of a professional engineer who specialises in software. A characterisation of this class of software might need to be added to the legislation.

As in all other areas of engineering, the body of knowledge and the principles of good practice (Engineering principles) for software engineering are evolving. As in each of the other areas the required body of knowledge will have to be regularly updated and the exams will have to be altered accordingly. Nonetheless, there is no reason why we cannot begin now.

Neither ACM nor IEEE should attempt to become a licensing authority. It would be