

From: Iwerth@ cs.utexas.edu

Subject: Establish Software Engineering As A Profession

We have received a couple of replies to our long memo that indicate that the reader believes that Laurie or I authored the Motion. In fact the motion is from Fletcher Buckley, a person from outside the ACM Ed Board, the IEEE-CS Educational Activities Board, the SEI or the executive committee of TCSE. In short it is an independent effort, but one with enough steam behind it that the education community needs to think about its response. Let us have your thoughts and your references.

Dear Educator,

The following motion or some slightly altered version of it will be discussed in several meetings at 1CSE (International Conference on Software Engineering) May 18-21, 1993. We will be attending many of these meetings and will have a chance to present information about the motion. There is a good chance that it will ultimately be considered by the IEEE-CS Board of Governors.

We would be interested in hearing your thoughts on the motion since it has many ramifications for educators.

For example:

1. Defining Software Engineering and Software Engineering Ethics

Do you think this is possible at this point? Do you consider these definitions to exist already (for example, in the SEI materials)?

2. Accrediting Software Engineering programs at universities

a. Is the Software Engineering curriculum mature enough for accreditation?

b. Are the universities and professional societies prepared to undertake accreditation in Software Engineering?

c. What would be the relationship between the existing ACM/IEEE-CS accreditation process for Computer Science, the existing ABET accreditation process for Computer Engineering and the proposed ABET Software Engineering accreditation?

Can the field and the universities support these subdivisions?

3. Encouraging the states to establish software engineering as a registered engineering field Are we ready for this'? What would be the impact on your program?

Laurie and John Werth Iwerth@cs.utexas.edu jwerth@cs.utexas.edu Vice-Chairs for Education Technical Committee on Software Engineering

Fletcher I Buckley
15 April 1993

MOTION

Move that the IEEE CS Board Of Governors appoint an ad hoc committee to initiate the actions to establish software engineering as a profession. This work should include:

- a. "Determining, in coordination with the Standards Activities Board, appropriate definitions and establishing those definitions as approved standards in accordance with IEEE Standards Board policies and procedures.
- b. "Determining, in coordination with the Educational Activities Board, the body of knowledge required for a four-year undergraduate curriculum for a Bachelor of Science in Software Engineering and establishing this as an approved curriculum at the Accreditation Board For Engineering Technology (ABET).
- c. "Determining, in coordination with the Membership Activities Board (**MAB**), a **set** of software engineering ethics.
- d. "Encouraging, in coordination with the MAB and the EAB, states to establish software engineering as a registered engineering field consistent with current practices in civil and electrical engineering."

BACKGROUND TO THE MOTION

1 HISTORICAL BACKGROUND

In 483 B.C., Xerxes, King of Persia and Media, as part of his campaign to conquer Greece, ordered two floating bridges to be constructed across the Hellespont to provide passage for his army from Asia to Europe. After the bridges were completed, a storm arose and the bridges were destroyed. Xerxes had the engineers killed and another set of bridges constructed, thus demonstrating at that time, the existence of standards of personal accountability for professionals working in their fields of their competence.

1.1 Current Situation

Today, we have a concept that holds professionals (doctors, lawyers, civil engineers, etc.) personally accountable for work in their fields. One major exception is software engineers. Today, we do not appear to have for software engineers:

- a. A defined body of knowledge that software engineers should have mastered.
- b. Any formal requirements for a technically-competent person to formally attest (certify) the validity of the design of critical software; that is, software whose failure would impact safety or cause significant social or financial losses.
- c. Any established criteria by which a judgment of the professional competence could be made of those individuals engaged in designing critical software.

1.2 Questions

In examining this state of affairs, several questions arise: a. Is there a professional field of technical expertise for software engineers?

- b. Is there a need to come to a judgment on the professional competence of those who develop and maintain critical software.
- c. Is there a need for standards of competence for those who design critical software'?
- d. How should we come to a judgment concerning the professional competence of such personnel'?

2 WHAT IS TO BE DONE

Examining the above, it appears that we need to do four things:

- a. Define the professional field.
- b. Identify the technical expertise required to be a member of that profession.
- c. Determine the entry-level requirements for the profession.
- d. Establish a common set of professional ethics.

2.1 Defining The Professional Field

The first step appears to identify a term that describes what is being done. This is not a trivial task as the definition of that term provides the basis for the remainder of the steps that follow. As in most efforts, a top-down approach appears to be productive and so that approach is taken here.

To design a system, we do system engineering. The system, in turn, is divided into hardware and software. To design the hardware portion, we do hardware engineering which, in turn, may include mechanical and electrical engineering. To design the software portion, we do software engineering which may, in turn, include database engineering and others.

The term "software engineering" also has a certain amount of existing stature:

- a. In the 1991 membership survey, over half (5470) of the current full members of the CS polled indicated that they considered themselves software engineers as did 40% of the affiliate members.
- b. The term used at this location is "software engineer" and I keep seeing that term in the "help-wanted" advertisements.

Assuming that we can use the term "software engineering" to describe what is being done, the next task appears to be to define that term. The IEEE definition is (1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.

(2) the study of approaches as in (1)."

That definition is currently under study to be modified and one proposal is that it be defined as follows:

"Software engineering: That branch of engineering that is concerned with the development, operation and maintenance of software."

This defines the term by identifying: (1) the next larger class (engineering) of which software engineering is an element, and, (2) the essential difference between software engineering and the other elements of that class.

Looking about for another definition has not yielded useful results. Consider, for example, the Final Report of the ACM Task Force on the Core of Computing Science. This identifies "software methodology and engineering" as one of the eight areas of "Computing As A Discipline". "Computing As A Discipline" appears to be defined as "the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation and application." There does not appear to be a simple definition of "software methodology and engineering", nor a definition of "software engineering" per se.

2.2 Identifying The Technical Expertise

The body of knowledge encompassed in other branches of engineering appears to be reasonably well-defined. Not so for software engineering, and the first question that arises is, can such a body of knowledge even be defined!

There is a group that holds the view that the body of knowledge for software engineering should be complete, and then and only then can we progress in defining the profession. However:

- a. From a historical view, the Romans were building roads, bridges and viaducts long before the mathematical foundations for stress/strain analyses became available. They did it heuristically; they knew what worked and extrapolated from there - and a number of those roads, bridges and viaducts are still in use today.
- b. Magnificent software systems (for example, airline reservation systems) have been built using the same heuristic techniques.
- c. From a pragmatic viewpoint, given that there is a need for software, what has to be done, will be done -- and it will be done with the tools and the knowledge available at that time.

2.3 Defining Entry-Level Requirements

Given that the field exists and that critical software is being built today, society, in order to protect itself in the use of a product, will eventually insist that persons who construct critical software have a modicum of competence in the field. This implies judgments to be made on what subjects should be mastered by those working in the field and it is at this point that good people can differ.

Simplifying the problem states that there should be a four-year undergraduate course in software engineering and not more than half the course should be subject specific; the rest is negotiable. To those who are concerned that more is needed and that those who come out of any such four-year curriculum will not be competent, there is only one response. No one who comes out of a four-year academic curriculum is competent. The only thing that they have gained is a solid foundation for entry into the field. There is a definite need for on-the-job training and there is no substitute for experience.

The determination of what a core curriculum of a four-year course in software engineering should be, is a difficult one, and one that will take considerable effort.

2.4 Determining Competence

In examining this situation, the next question that arises is: how should we go about determining the competence of those building critical software? Looking for analogies in the hardware field, we find state boards both:

- a. Requiring that engineering designs be certified by registered professional engineers, and,
- b. Establishing requirements for registration in a specific engineering specialty. These requirements usually include graduation from a specified curriculum at an accredited engineering school, passage of one or more examinations in the specialty area, and a specified amount of time in the practice of the profession.

Now there are many unanswered questions about this topic, but one thing appears certain: sooner or later, the states, the national governments, and/or the European Community is going to require that the design of critical software be certified by those already judged to be competent to do so. So either we must establish appropriate machinery to judge competence in the software engineering field, or it will be done for us, by others.

2.5 Establishing Professional Ethics

The ethical aspects of our work have been explored in two previous issues of the Standards Department (March '90 and Feb '91), and Section 7.8 of the IEEE Policies and Procedures Manual could perhaps provide an initial starting point. But the issues associated with ethics are too important to expect that an appointed committee's work could represent a consensus of the concerned professionals in the field.

3 INITIATING THE PROCESS

3.1 Overview

To obtain a consensus of the concerned professionals in the field on the norms of the Software engineering profession, it appears that we need to form a group to formulate these norms. A, a first reasonable approach, consider that:

a. Participation should be open on an equal basis to all persons who are directly and materially affected. Further, we should actively solicit membership both inside and outside the Computer Society.

It is not clear that we should solicit organizational memberships for this group. Rather, it appears preferable that members should represent themselves as concerned professionals in the field.

If organizations desire to send an organizational representative, they should be accommodated, and their inputs valued and processed as would be any other input. However, no organization should be given a veto power over results that reflect the consensus of the concerned professionals in the field.

b. The group should not be dominated by any single interest category.

c. The conduct of the group should be governed by written procedures.

d. Prompt consideration should be given to the written views and objections of all participants and that each objector should be notified of the disposition of an objection and the reasons therefore.

e. Unresolved objections and the associated reasons should be reported to the entire membership.

f. The conclusions of the group should be supported by a consensus of the membership. This would imply a ballot on the final products, probably run as draft standards ballots are currently run.

3.2 Initial Schedule

A tentative schedule is as follow:

a. March '93 - April '93: Coordinate and revise this proposal based on the comments received.

b. May '93: Gather in the collected thoughts of the CS BoG and its associated boards at the May '93 meeting. As part of this, I would ask the Chairs of the EAB, MAB, SAB, TAB, and COPP to reserve a time-slice on their agenda for a short presentation and comments on this topic.

c. June '93 - August '93: Publicize the effort in appropriate magazines (for example, COMPUTER, COMMUNICATIONS OF THE ACM, TCSE Newsletter, SigSoft News, Software, and Crosswllk). Further suggestions would be appreciated.

d. November '93: Hold first meeting of the group during the week of 8-12 November 1993 concurrently with the BoG meetings at Santa Clara, CA. This is projected to a 1 1/2 day meeting beginning at 8:00 am on Monday, 9 November 1993 and ending at noon, Tuesday 9, November 1993 This is projected to be hosted by the CS at the hotel where the BoG meetings are being held. Further milestone, to be projected a, a result of the May -- October efforts.

4 ORGANIZATIONAL ASPECTS

Determining the organizational sponsor for this effort includes the following consideration:

a. One thought that has been strongly considered is that it should be a task force of the TC on Software Engineering. The scope of the effort appears to be across so many different CS BoG Boards, however, that it is not certain that such a placement would be correct:

(1) There is a major focus in the Educational Activities Board for the definition of the curriculum for a four-year undergraduate degree and its establishment at ABET. This also involves the IEEE EAB's Accreditation Policy Committee.

(2) The Membership Activities Board's Committee On Public Policy has a major role to play, particularly in the formulation of software engineering ethics and from their activity in software safety.

(3) Both the MAB and the EAB have a strong interest in registration as professional software engineers at the state level. This action also involves the USAB's Licensure and Registration Committee.

(4) The Standards Activities Board has been establishing standards for software engineering since 1976 and would be directly involved in the establishment and modification of definitions, among other interests.

(5) The Intersociety Cooperation Committee should be involved as this will go well beyond just CS interests and activities.

(6) There may very well be other CS interests that could contribute and should be involved. (Your help in identifying these would be appreciated.)

b. In view of the above, and recognizing that the desire to accomplish a specific purpose, it appears that a better approach would be the appointment by the BoG of a special (ad hoc) committee to carry out at least the initial effort.

5 FINANCIAL IMPACT

1993 Computer Society support requirements are projected as follows:

- a. Provide a meeting room at the November '93 BoG meeting to take place as indicated above.
- b. Provide support for mailings to group members.
- c. Provide room for an announcement in the July '93 or Aug '93 Computer.

This support is considered to be estimated at less than \$2,000 and will be requested to be provided by the President of the CS under his authority per section 17.2.3 of the CS PPM.

In a similar manner (assuming that the work shows sufficient promise that it goes on into 1994), Computer Society support requirements for 1994 are projected for three sets of meetings (at the BoG meeting weeks) and associated mailings, and are estimated at \$6,000.

Should this effort prove to be a burden, then a nominal assessment will be considered to cover the cost of mailings and meetings. This might be in the range of \$35.00 per year. This will come more sharply into focus at the November '93 ad hoc committee meeting.

6 ADDITIONAL ITEMS

The following have been brought to my attention since the initial draft of this paper.

6.1 References

- I. Herodotus, "The Histories", Penguin Books, Ltd, 1954, p 429.
- 2. "Survey Of Members, Former Members, And Potential Members For The Computer Society Of IEEE", Washington, DC., Laurence Leiter And Company, May 1991, pp 13, 21.
- 3. ANSI/IEEE Std 610.12-1990, IEEE Standard Glossary Of Software Engineering Terminology.
- 4. Communications of the ACM, Jan 1989, pp 9-23.
 - a. Ford, G., "SEI Report On Undergraduate Software Engineering Education", CMU/SEI -90-TR3, ADA223881.
 - b. Parnas, D., "Education For Computing Professionals", Computer, pp 17-22. Comments by McGonnigal, et al, Computer, Apr 90, pp 8-9.

6.2 Information

Contact should be made with:

- a. The ICCP. They have worked on parts of this topic for some time now.
- b. The American Society For Quality Control (ASQC). Their software division has some activity in the certification field.
- c. The British Computer Society. They have a certification program and have published job-oriented descriptions.