To: Rachelle Hollander

From: Vivian Weil and Michael Davis

Date: April 12, 1995

Subject: Addition (Codicil) to Proposal: What Balance Between Technical Standards, Professional Standards, and Professional Education

---

Why have you selected that group (the Task Force on Software Engineering Ethics and Professional Practices)?

One reason, of course, is that its domain corresponds most closely to the long-term work of CSEP (and the expertise of the principal investigators). We are an ethics center, not (like Carnegie-Mellon's Institute for Software Engineering) a center concerned with purely technical standards. Of the three task forces, we are most likely to be both effective researchers and effective participants in this one. We have chosen the task force that speaks our language.

A second reason is that the cooperation of two powerful professions (electrical engineering and computer science) creates a field with fewer settled questions than is usually the case when a new profession (say, environmental engineering) simply branches off. Engineers and computer scientists are similar enough to understand each other, yet different enough to require an explicitness about professional matters not necessary within a profession unified by history.

What models for technical standards and ethical standards will prevail? On the one hand, engineering's longer history and its traditional concern with safety might give its standards overwhelming weight. On the other hand, the ACM has included "Organizational Leadership Imperatives" in its code, reflecting an approach that puts the organizational context of practice at less distance than that of the engineers' codes. The SEEPP effort may give us evidence of whether an approach that brings employers and even customers into the picture is gaining ground. (See SEI's Capability Maturity Model.)

At present software developers lacking an ABET-approved engineering degree are barred in at least 48 states from presenting themselves to the public as software engineers. This is a refusal to recognize software engineering degrees awarded by computer science departments. An interpretation in terms of the received wisdom about professions is that this is a turf issue. Yet it might reflect a genuine disagreement with regard to standards or education. Will members of the task force be surprised from time to time by the standards members of the other profession take for granted? Will the computer science people take pains to appropriate the presumed sophistication of technical engineering standards? Will there be significant disagreement requiring accommodation on both sides with regard to education, standards, or both?

A third reason is that we are in a position to confirm, or disconfirm, certain views of professions, especially of their codes of ethics, common in the literature of professions. For example, it is common to claim that codes are meant to serve the profession at the expense of the public, that they are designed not to be followed but to hoodwink the public, that they are meant to protect the powerful within the profession from the upstarts, or that they are a conspiracy against an otherwise fair market. Such claims, common in the sociology of professions, do not seem to rest on a careful study of the process by which professions in fact come into existence (in part, no doubt, because that process is seldom well-documented). [Cf. Margali Larson, etc.]

We have an opportunity to study how the required professional education of software developers is determined. The received account depicts the professions seeking status within universities and constructing their bodies of knowledge as the "science" on which the profession depends. In this instance, the actual discussion can enlighten us about how the training is conceived in relation to the demands of practice and of the marketplace. Just as important, it can reveal the degree to which ethical considerations inform the determination about what the training should include, in either computer science or electrical engineering departments.

Is this effort likely to tell us something important about professions in relationship to society?

Our hypothesis is that most arguments, especially those generally treated as legitimate, will fall into one the following categories: a) those that appeal to what's practical (what can you really expect people like us to do?); b) those that appeal to the public welfare (or, at least, the welfare of employers and users); c) those that appeal to what the public (or, at least employers and users) want (whether it's good for them or not); and d) those concerned with consistency between software developers and those with whom they work or want to be allied (for example, the standards of the ACM or IEEE). Davis has, of course, long argued for something like this interpretation of codes. But, until now, he has had to do so using the codes themselves, explaining what they actually say in less jaundiced terms than has been common or pointing to the direction

of change over time. That, of course, is not nearly so satisfactory a way to counter the common view as simply saying, "Here's the debate: where are the selfish interests?"

Our method will be a) to identify (kinds of) argument made to justify shaping the standards this way rather that and b) to gauge what part they play in the process (e.g., how often they are made, how often treated as decisive, respectable, woolly, or beyond-the-pale). We will, in short, be doing simple content analysis.

While we do have a hypothesis, we <u>are</u> on a "fishing expedition"--and should be proud of it. Our chosen task force has had a slow start (despite the co-chairs' efforts). We are already wondering whether starting a new profession, even with adequate resources, might be considerably harder than the standard literature led us to expect. That is something we will want to pay attention to. We hope for more such surprises.