

Chapter 3: SEEPP Begins, 1994

“Nothing is as easy as it looks.”
—The First of Murphy’s Laws

3.1 The Steering Committee

By March 4, 1994, the time at which the IEEE-CS Steering Committee for the Establishment of Software Engineering as a Profession presented its final report to the IEEE-CS Board of Governors, the new IEEE-CS/ACM Joint Steering Committee had already been working for at least three months. Indeed, the handing off of work from one committee to the other was so smooth that there does not seem to be a day on which it can be said to have occurred. We can be sure that it was complete by March 4 because Mario Barbacci, chair of both committees, had presented the first report of the new Joint Steering Committee to the ACM a few days before (in Nashville, Tennessee). By then, he had a new vice-chair from the ACM, Dennis Frailey,¹ and a much enlarged committee, all members but one identified in the report with an “A” (for ACM) or “C” (for IEEE-CS). The one exception, Patricia Douglas, member ex officio because she chaired the body-of-knowledge task force, was a psychologist at IBM Skill Dynamics. Though in fact a member of IEEE, she seems to have been treated as neutral. One or both of the following considerations explain her position: first, she seems to have had the trust of both the ACM and IEEE-CS; second, there was no one at the ACM (or anyone else at IEEE-CS) who wanted to co-chair her committee. Defining the body of knowledge was a daunting task best left to experts.

Six of the seven IEEE-CS members should be familiar from Chapter 1. Barbacci himself, Buckley, Chikofsky, and Druffel were from the original committee. Two of the ex officio members, Carver (Curriculum) and Melford (SEEPP), were from old IEEE-CS task forces (now IEEE-CS/ACM task forces). The one new IEEE-CS face, an “ex officio”, Ron Hoelzeman, is an academic (Electrical Engineering-Computer Engineering, University of Pittsburgh), apparently a replacement for Schneidewind.²

Three of the seven ACM members are also familiar from Chapter 2.3. Zweben had served as the ACM representative to the old IEEE-CS Steering Committee. John Werth,³ the new co-chair of the curriculum task force, had been (and was still) one of the Vice Chairs for Education of the *IEEE* Technical Council on Software Engineering (TCSE).⁴ Gotterbarn, now co-chair of the ethics task force, had been active in earlier IEEE-CS discussions of licensing.⁵ Mary Shaw is familiar for another reason, her extended critique of Buckley’s original proposal at the TCSE meeting (2.4). The two other ACM appointees—Barry Boehm and Stuart Feldman—are new. Both would have carried considerable authority with other ACM members.

Barry Boehm holds a B.A. (Harvard, 1957), M.A. (UCLA, 1961), and Ph.D. (UCLA, 1964), all in math. He had worked at TRW from 1973 to 1989, ending his career there as Chief Scientist of the Defense Systems Group. Between 1989 and 1992, he was at the U.S. Department of Defense, serving as Director of the Information Science and Technology Office, Defense Advanced Research Projects Agency (DARPA), and as Director of the Software and Computer Technology Office, Defense Department Research and Engineering (DDR&E). Since autumn 1992, he had been TRW Professor of Software Engineering, Computer Science Department,

University of Southern California. He had authored several books, including *Software Engineering Economics* (Prentice Hall, 1982), *Controlling Software Projects: Management, Measurement, and Estimates* (Prentice Hall, 1986), and *Software Risk Management* (IEEE Computer Society Press, 1990), all then still in print.⁶

Feldman was at Bell Communications Research, Inc. (“Bellcore”, now Telcordia Technologies) in Livingston, New Jersey. He held a management position in software engineering and computing systems and was the Technical Leader of the Telecommunications Information Networking Architecture Consortium, an international research group of telecommunications and computing companies. Feldman held an A.B. in Astrophysical Sciences and Mathematics (Princeton, 1968) and a Ph.D. in Applied Mathematics (MIT, 1973).⁷

As we look over this list (ignoring Douglas), we can see an obvious pattern. The Steering Committee is joint in a more complicated way than the official rationale admits—and divided in a way never officially stated. On the one hand, all the ACM members are (as a matter of fact) also members of IEEE-CS. This is not as surprising as it may seem. The membership overlap was (and remains) considerable. Some estimates are that 90% of ACM members are also members (or affiliates) of IEEE-CS (with only a somewhat smaller percentage running the other direction because the IEEE-CS is somewhat larger).⁸ What we have, then, is not a distinction between IEEE-CS members of the Joint Steering Committee and ACM members but merely between IEEE-CS *appointees* and ACM *appointees*. What explains the overlap? Or, to put the question another way, what explains the continuing existence of *both* ACM *and* IEEE-CS? Why have they not long since merged?

The overlapping membership seems to have a relatively pedestrian explanation. Neither IEEE-CS nor ACM is a professional society (strictly so called) but a scientific or technical society, that is, an organization (formally) held together by interest in a field of knowledge (rather than, as a professional society would be, by occupation). To join IEEE-CS, one has to meet only *one* of the three criteria: a serious interest in any aspect of the computer field, membership in an affiliate society, or membership in IEEE or another IEEE society. Of the *three*, the hardest to satisfy may seem to be IEEE membership. But even that was not (and is not) hard for most ACM members. To join IEEE (as a “member”, as opposed to an “affiliate”)⁹, one only needed a bachelor’s degree from an ABET-accredited program *or a CSAB-accredited program* (or, absent such a degree, what the admissions committee would count as three years of experience in a position normally requiring such a bachelor’s degree).¹⁰ To join the ACM, one needed only a bachelor’s degree (in any field), the equivalent level of education, or two years full-time employment in information technology.¹¹ Most members of ACM would have had more than two years full-time employment in information technology (in a position normally requiring a bachelor’s degree), making them eligible for membership in IEEE (and so in IEEE-CS).

Why are there two organizations rather than one? The answer seems to be that, on the whole, they do different things. IEEE-CS shares with the rest of the IEEE a set of practical concerns characteristic of engineering, especially development of technical standards. The ACM, in contrast, is more concerned with the science of computing (especially, the mathematics of computing) and with the computer science curriculum. The two associations hold different sorts of conferences, publish different sorts of journals, and otherwise work side by side rather than in direct competition. *Collected Algorithms* or *Transactions of Programming Languages and*

Systems is a typical ACM journal; the IEEE-CS equivalent would be *Computer or Transactions on Mobile Computing*.

The explanation of this cohabitation is probably history rather than logic. The two associations began about the same time, one as a special interest group within an engineering society and the other as an independent society concerned more with the *applications* of computing machinery (what we now call “programs” or “software”) than with the design of the machines themselves (engineering). As software became an ever larger part of the “machine”, the distinction between engineering and applications became ever harder to make. Indeed, the term “software engineering” may be understood as a declaration that applications were as much in need of engineering as the hardware was. Nonetheless, there remain significant differences between the two organizations. Perhaps the most important is that the ACM has always had a stronger identification with student organizations in departments of computer science than IEEE-CS has.

3.2 The Odd Couple: Melford and Gotterbarn

These differences between the two societies may explain an important difference between ACM appointees and IEEE-CS appointees. All but one of the IEEE-CS appointees (Melford) have at least one (ABET-accredited) degree in engineering (or the non-US equivalent). None of the ACM appointees does. In contrast, all but one of the ACM appointees (Gotterbarn) have at least one degree in mathematics. The exceptions confirm what we know about the flexibility of the formal membership requirements of the two organizations.

Melford, a member of both organizations, was not an engineer; indeed, he was not even a college graduate. (When he left California State University at Irvine two years before graduation, his major was economics, though he had planned to study electrical engineering.)¹² He joined the IEEE in the early 1980s because a colleague encouraged him to. (By then, he had three years experience in security applications of software that would ordinarily be done by people with a bachelor’s degree in computer science or engineering; hence, there was no problem about his membership even though he was not trained as an engineer.) Some years later he got a broadcast mailing from the IEEE-CS saying that the Committee on Public Policy (COPP) had started an ethics subcommittee and asking if he was interested in joining it. Because he had concluded that most problems with software were ultimately problems with those who wrote software, he joined. By the time the Steering Committee was setting up the task forces, Melford was COPP’s chair, an obvious choice for IEEE-CS co-chair of the ethics task force.¹³

Donald Gotterbarn, the ACM co-chair, differs from Melford in at least three important ways. First is education. Gotterbarn had a lot more of it (and what he had was quite different from Melford’s). Gotterbarn holds a B.A. in philosophy from Hofstra University (1964); a Master of Divinity from Colgate Rochester Divinity School (1967); and an M.A. (1970) and Ph.D. (1971), both in philosophy, from the University of Rochester. (His dissertation’s title is *Natural and Philosophical Relations in David Hume*.) He taught philosophy at Wichita State University (Kansas) for three years (1971-74), turning down early tenure to accept an untenured assistant professorship at the University of Southern California’s School of Philosophy. From the perspective of most philosophers, to teach at WSU is to toil in the lower reaches of academy’s Purgatory; to move from there to USC, to rocket into its Heaven. So, what happened two years

later should surprise. In 1976, Gotterbarn left USC for Harrisburg, Pennsylvania, without even a job in hand.

There were several reasons for leaving USC just two years after he arrived. Both Gotterbarn and his wife were Easterners much farther from relatives than they had been in Kansas. Though they had grown up in New York City, they had developed a strong dislike for Los Angeles. These reasons, and perhaps others, led to the sudden re-direction of Gotterbarn's career.

For the Spring Term of 1977 and again for the 1979-80 academic year, Gotterbarn taught philosophy part time at Dickenson College (in nearby Carlisle), more because he liked teaching than because he needed the money. He made his living as a programmer.

Gotterbarn's preparation for programming was similar to Melford's, that is, he was (more or less) self-taught.¹⁴ Having learned symbolic logic as part of his training in philosophy, he supplemented what he could teach himself from books on programming by sitting in on computer science courses at WSU and USC. To improve his job prospects in Harrisburg, he entered a two-year program in computing at Harrisburg Area Community College, graduating with an Associate of Arts (A.A.) in computing in 1979. (At the time, the A.A. was a recognized credential in applied computing.)

A second difference between Melford's background and Gotterbarn's is their respective careers within computing. Gotterbarn's early computing career is similar to Melford's (and was roughly contemporaneous with it). From 1976 through 1984, Gotterbarn worked as a programmer, moving quickly from simple programming to systems development. By the early 1980s, he had his own consulting firm. As a consultant, he did work on Medicare programs for Blue Cross Pennsylvania, helped Blue Cross develop networks between several of its regional operations, and developed inventory tracking systems for the US government's Ship Parts Control Center in nearby Mechanicsburg and a similar facility in Philadelphia.

But, for all this practical experience, Gotterbarn seems to have remained an academic. While there is a gap of fifteen years in the *Philosopher's Index* between his last publication concerned with Hume (1976) and his first publication concerned with computer ethics (1991), those years were not without publications. Gotterbarn's curriculum vitae lists (in addition to what the *Philosopher's Index* has for these years) the following: "James' Theory of Cognitive Knowledge", *Bicentennial Symposium* (1976); "Tipton on Berkeley's Immaterialism", *Philosophia* (1979); and nine entries for the *American Academic Encyclopedia* (1980).¹⁵

For the first few years after leaving USC, Gotterbarn seems to have tried to keep up with developments in his old field, the history of modern philosophy. But, more important, he began to identify ethical questions in the programming he was doing. The first such question concerned the transfer of some medical decisions until recently made by physicians, nurses, and other "health care providers" with direct knowledge of the patient, to computers or to distant Blue Cross employees with a computer assistant (an expert system). What responsibilities did programmers who designed or built the necessary software have as a result of this transfer of decision-making to people unable to see the consequences of their own decisions or to evaluate the assumptions on which the software was constructed? Did the programmers have the same responsibilities to patients as physicians did? Or greater responsibilities because their software had a much larger impact on patients than the decisions of any single physician could have?

Gotterbarn had also begun to work part time as a "faculty tutor" in the computer science lab at Harrisburg Area Community College.¹⁶ Some of his students might (he realized) soon be

helping to design and build nuclear weapons. Was it right to teach them the programming skills they would need to do that? Then, in 1983, President Reagan announced the Space Defense Initiative (a space-based defense against nuclear missile attack heavily dependent on software of a scale, complexity, and reliability well beyond anything yet attempted). Here was another set of questions to consider. Many programmers doubted anything like the specified system could be built for several decades, if at all. They were also concerned that any system much like that proposed could only be trusted if it could be adequately tested under realistic conditions and such testing was not practical. (Who would want nuclear explosions going off a hundred miles or so above their heads?) Should programmers nonetheless work on such a project, since the technical problems were interesting and the money was good? What responsibility did programmers have to tell the truth about what they did, to warn clients and public about the risks inherent in a program too complex for anyone to understand, about the limits of what could be accomplished?

The similarity between Gotterbarn's career and Melford's ends here. By 1983, Gotterbarn was tired of consulting. (For him, too much of consulting was winning the contracts to do the work; his interest was the work itself.) So, in early 1984, he took a salaried position at a Scranton bank, designing electronic fund-transfer systems. (Scranton is about 120 miles north of Harrisburg.) When, a few months later, the bank was taken over by a larger bank uninterested in computer research, Gotterbarn turned down its offer of a position managing programmers to become a (much-lower-paid) assistant professor of computer science at Alleghany College (in Meadville, Pennsylvania, almost 300 miles west of Harrisburg). There he taught basic computer science courses and developed some new courses. Most of the new courses, such as one in database design, were technical. But one, a course in computer ethics, combined the technical with the ethical. Students actually developed software and reviewed technical documents looking for technical as well as ethical errors.

Gotterbarn was particularly well-suited to teach computer ethics. For many professors of computer science then, a course in computer ethics was not a computer science course at all. For them, computer science was a kind of mathematics, very far from practice. Computer ethics was a kind of ethics. Philosophers taught ethics. Gotterbarn's Ph.D. in philosophy thus seemed to make him the ideal person to teach the course. Gotterbarn also thought himself the ideal person to teach the course, but not (primarily) because of his training in philosophy. Unlike his colleagues, he had done programming "in the real world". He had, he liked to say, written programs that, written incorrectly, could kill people. He could convey to his students the uncomfortable truth that programming was not mere mathematics but a practical activity capable of doing great harm as well as great good. Part of avoiding harm was writing programs that met certain standards, standards of ethical conduct as well as of competent programming. Indeed, competent program was an ethical imperative (as well as a technical desideratum).

Courses in computer ethics were still rare enough then to be news. In 1988, the *Chronicle of Higher Education* carried a story entitled "Failure of Colleges to Teach Computer Ethics is Called an Oversight with Potentially Catastrophic Consequences". Gotterbarn was quoted in the article—along with another philosopher, Deborah Johnson (whose textbook *Computer Ethics* was described as one of the few in the field).¹⁷

During the years in Harrisburg, Gotterbarn had not had time to put his thoughts about computer ethics on paper. He had been doing too much else. Now, an academic again, he began to write—and to read what he was writing at conferences. The first of what would soon be many

papers was “The Roles of Computer Ethics in the Curriculum”, presented at the Conference on Computing and Undergraduate Education held at Carnegie Mellon University (1987).

Meadville is only 90 miles north of Pittsburgh where (as explained in Chapter 2.1) the Software Engineering Institute (SEI) opened on January 1, 1985. Gotterbarn was soon attending SEI seminars and conferences. Some of SEI’s first initiatives were educational. These included an attempt both to define a curriculum for software engineering and to raise the technical competence of software engineers already in practice. These were major undertakings. (At the time, as Gotterbarn recalls, the chair of his computer science department liked to say, “Software engineering is just about documentation.”). Gotterbarn approved of both initiatives. By late 1987, he was inquiring about participation in some of SEI’s educational initiatives.

In 1988, Gotterbarn went back to Wichita State, becoming an assistant professor of computer science. The move is not as surprising as it may appear. WSU’s department was the first to experiment with a Masters of Software Engineering using a curriculum SEI was developing. WSU thus gave Gotterbarn the chance to cooperate with SEI in the development of courses, to test them, and to help SEI draw from the results educational materials that could be distributed to other universities trying to establish similar programs in software engineering.

Early in 1989, Gotterbarn presented a paper on ethics at SEI and lobbied SEI staff to include ethics in the new curriculum. The lobbying proved effective. Gotterbarn was named a Visiting Scientist at SEI for the summer and asked to develop educational materials to be used in software engineering classes, including some materials on ethics. It was then, Gotterbarn believes, that he coined the term “software engineering ethics” to describe the ethical responsibility of software developers in the practice of their craft.¹⁸ In any case, he began the summer with the first Software Engineering Ethics Workshop. It was there that he met computer scientists, Keith Miller (then at the College of William and Mary in Virginia) and Joyce Currie Little (Towson University, Maryland), both of whom had undergraduate degrees in (math) education (as well as Ph.D.’s in computer science). He would later recruit both for SEEPP.

By fall 1989, Gotterbarn knew a good deal more about WSU’s computer science department than he had a year earlier. It was a less happy organization than it had seemed at first and, because of a new chair, less committed to software engineering. Gotterbarn therefore began looking for another job. He soon found that East Tennessee State University (ETSU) was looking for someone to teach software engineering courses and to start a master’s degree in software engineering following the SEI model.

ETSU had other attractions to compensate for its isolation in northeastern Tennessee. Johnson City, with a population of about 60,000, was the size that Gotterbarn had found he liked. The location, a long valley, with the blue-ridged foothills of the Great Smokey Mountains rising abruptly on either side, was at least as beautiful as anything he had seen in Pennsylvania—and quite unlike the plains of Kansas (or the smoggy hills of Los Angeles). The university was of medium size (about 10,000 students) with a spacious campus of grass, trees, and streams. Its buildings spread out as if land were plentiful and cheap. Most of the buildings were of red brick with white trim in one approximation or another of Georgian colonial, giving the place an eastern look. Gilbert Hall, one of the oldest buildings on campus, was where Gotterbarn would have an office on the second floor, along with the rest of Computer Science (Mathematics on the floor below). Almost a century old, with an impressive façade, Gilbert’s interior was nonetheless utilitarian and undistinguished. Of course, the chief attraction of ETSU’s computer science department was not its offices. ETSU’s department was “applications oriented”, indeed, so

applications oriented that it already had a course in computer ethics. Gotterbarn did not have to explain to that department why it was important to teach computer ethics—or that a course in computer ethics was a computer course, not a philosophy course. He arrived in the fall of 1990 as an associate professor, moved into a windowless office, and has taught there ever since—except for an occasional visiting professorship elsewhere.

It was also about this time that Gotterbarn became a “Distinguished National ACM Lecturer” on ethical issues in computing (as well as on fiber optic computers and new developments in telecommunications).¹⁹ Gotterbarn was developing a reputation among ACM members interested in computer ethics.

3.3 The ACM Code

A third difference between Melford and Gotterbarn is that Gotterbarn had had some experience in writing a code of ethics before joining SEEPP, while Melford had none. Gotterbarn had helped to write the present ACM code of ethics. In 1991, Ronald E. Anderson, a former president of the ACM Special Interest Group in Computers and Society (SIGCAS), received a grant from ACM to revise its Code of Professional Conduct, adopted in 1972.²⁰ This was the first official endorsement of a project Anderson had already been actively working on for at least a year. He had begun that work at an all-day symposium at the first Computers and Quality of Life Conference (1990). Anderson presented ideas derived from that meeting at other meetings.²¹ Now, grant in hand, he recruited from attendees at the annual Computers and Society Conference (February 1991) a dozen people to produce a draft. Among those recruited, beside Gotterbarn, were two professors of computer science whom Gotterbarn would later recruit as well (Dianne Martin and Laurie Werth).²²

Over the next five months, Anderson prepared a “rough draft”—which he distributed at the Conference on Computing and Human Values, sponsored by the National Science Foundation and hosted by the Research Center for Computing and Society (RCCS) at Southern Connecticut State University (New Haven), in August, 1991. He also sent a copy of this draft to the ACM Council (the ACM’s governing body). The Council’s reaction illustrates how careful a drafting committee should be about who sees its early work—or, at least, about how it is presented. The Council’s reaction to Anderson’s rough draft was not good. The primary reason (he soon learned) was that Council members supposed that the rough draft was the final draft. Once that misunderstanding was cleared up, there remained a number of substantive criticisms.

At the Conference on Computing and Human Values, Anderson found the people he needed (about fifty) and the free late afternoons and early evenings to rework the rough draft. He presented the rough draft along with the criticisms he had collected. By the end of the conference, he had a presentable “first draft”.²³

Anderson distributed this official draft to the ACM Council. The Council responded. Anderson’s drafting committee then revised accordingly (working in part at least by email). He sent the Council the revised code and received further responses which he tried to incorporate as he had before. Anderson was also presenting the draft at various ACM-sponsored meetings, collating responses from his audiences. Eventually, Anderson had what he supposed a final draft.

The process of creating a code of ethics, especially winning adoption, is always “political” (that is, an activity requiring participants to work together voluntarily, winning by persuasion what cannot be compelled). Anything from objective considerations (whom the code

would help or hurt) to personal taste in language might decide the outcome. As the drafting committee came to appreciate who wanted what and why, it revised the code accordingly. So, for example, Imperative 4.1 now simply urges ACM members to "Be fair and take action not to discriminate". Originally, it included a short list of prohibited subjects of discrimination, but various special interest groups wanted to add their own favorite to the list. To avoid a long list, the drafting committee decided to make it clear that the list was not intended to be exhaustive. The committee did that by moving the entire list from the Imperative to the corresponding Guideline. The generality of the Imperative now captures all types of discrimination. The Guideline provides enough specificity ("race, sex, religion, age, disability, national origin, or other such similar factors").²⁴

The draft that emerged from this process was published in *Communications of the ACM* (May 1992) with a ballot asking members to vote on each item in the code. The vote was favorable for all provisions—clearing the way for the final Council approval. Then, a few days before the Council's meeting, Anderson learned that some Council members might vote against the new code because they believed it would not be useful in decision-making. (The Council included some members who had helped to write the code this new one was to replace.) Anderson asked Gotterbarn and Deborah Johnson to help write a paper showing the new code's usefulness in ethical decision-making. (By then, Johnson's text in computer ethics was widely used.) The three quickly wrote the paper, "Using the ACM Code of Ethics in Decision Making", and presented it at a Council meeting the morning of October 16, 1992.²⁵

Since the presentation went well, neither they nor Anderson stayed for the afternoon session at which the code was to be approved. That may have been a mistake. The Council made a number of revisions before coming to a final vote. One of those revisions was significant (weakening the clause providing for penalties for disobeying the code). To this day, Gotterbarn believes that the drafting committee could have prevented those revisions had it been there to explain why it had written the code the way it had. One lesson Gotterbarn took from that experience is that those who know most about a proposed code should be present at deliberations concerning it until final approval. One cannot expect those who have not worked on the drafts and participated in extended discussions about them to understand the final draft as well as those who did.²⁶ Another lesson he might have drawn from his experience with writing the ACM code was that even a large organization could write a code of ethics in eighteen months.

3.4 SEEPP Gets Started

Gotterbarn first met Barbacci on November 16, 1993.²⁷ Gotterbarn was at SEI on business unrelated to the Steering Committee's work. In the morning, he video-taped a graduate class on professional issues in software management. (By then, SEI had a substantial distance education program.) In the afternoon, he made a CD for SEI's "Just in time education program". (SEI could generate CDs quickly in response to hot topics, marketing them to industry.) Between these activities, over lunch, he met with Barbacci. Barbacci believes Gotterbarn arranged the meeting.²⁸ (Gotterbarn does not remember.) Its official purpose was to "discuss" the IEEE-CS/ACM joint project. Barbacci sketched the plan so far developed, gave Gotterbarn copies of the minutes of the relevant meetings of the IEEE-CS Board of Governors and ACM Council, and asked about his interests, experience, and opinions. Barbacci also asked whether Gotterbarn was interested in participating in the project. He answered that he was.

A few days after the meeting, Gotterbarn decided to put his thoughts in better order. The result was a three-page single-spaced memo commenting on specific passages in the documents he had received. To this he added a copy of a paper he had written on licensing and emailed the package to Barbacci. The gist of the paper was that licensing was not an end in itself but a way to enforce standards, especially, standards of skill. The gist of the memo was that neither the IEEE-CS nor the ACM seemed to have a good grasp of what goes into making a profession; they certainly did not agree with Buckley that software engineering should be a profession. Getting agreement on significant standards was worthwhile, but significant standards had to be detailed, more detailed than either the IEEE-CS or ACM document suggested. Gotterbarn seemed to be advocating (what he would later describe as) “standards of practice” rather than a “mere” code of ethics. Neither the memo nor the accompanying paper has much to say about ethics. Indeed, the memo ends with a postscript asking, “How does ICCP [Institute for Certification of Computer Professionals] fit into the effort to define skills?”²⁹ Nowhere in the memo does Gotterbarn express any idea that he might co-chair one of the task forces (though he does thank Mario for “giving me the opportunity to participate”).

Nonetheless, Gotterbarn was soon offered the opportunity to co-chair the ethics task force; he accepted; and, on January 29, 1994, Barbacci met with him and Melford in Pittsburgh to get the task force started. The meeting took place in the same SEI seminar room in which Gotterbarn had held his first workshop on software engineering ethics (1989). There was an irony in this. By 1992, thinking about ethics had changed at SEI. Norman Gibbs, then the Director of SEI’s Education Division, had told Gotterbarn, “Ethics is not relevant to software engineering” and, proving he meant what he said, announced that SEI would not be publishing the curriculum module on ethics Gotterbarn had prepared. Now, less than two years later, ethics was back at SEI (if only on a weekend and only because Barbacci had his office there).³⁰

This was when Gotterbarn and Melford first met. There was a long agenda. The meeting went well. The three began by refining (or broadening) the task force’s assignment. The task force had been told (two months before) to develop a “code of ethics”. The January minutes (Melford’s) announce that “Codes of Ethics [are] (not equal to) Standards of Conduct”, distinguishing them in this way: A code of ethics merely “identifies conflicts”. Standards of conduct provide: “a) Guidance to resolve conflicts [between users, developers, employers, regulators, and others], b) Standards of quality, and c) Responsibility for understanding the implications of one’s work.” This distinction seems to explain the name by which the minutes refer to the task force for the first time, something that seems to need explanation (in part, at least, because the Steering Committee had created a task force to write a “code of ethics” and said nothing about “a code of professional practice”). The three had to reject (or overlook) the obvious name for the task force, the “Software Engineering Ethics Task Force”, with its pleasant acronym, “SEE”, for “Software Engineering Ethics *and Professional Practices*”, with the unpleasant hint of ooze in the acronym “SEEPP”. The three were unhappy enough with “SEEPP” that they agreed to find a better name “later”. But events preempted good intentions. SEEPP had to use its name publicly before a better one could be found—and then a better would have had to be bought at the price of some confusion.³¹

The term “professional practices” suggests standards of conduct, something more than just “ethics”. In choosing to include “professional practices” in the task force name, the three may seem to have been committing SEEPP to report something more than a code of ethics, that is, a “code of practice”. There is, however, another interpretation—for Gotterbarn, at least. The

three-page memo he had sent Barbacci after their first meeting (November 16) included the following comment (concerning “page two”):

In general, I would suggest keeping constantly coupled the phrases “ethical conduct and professional practice.” The definitions of professional you included do not indicate the contractual nature of a profession. The theoretical contract of professionalism says that “by virtue of having a specialized skill and being allowed by society to maintain control over this skill, society is due a HIGHER standard of care from the (licensed) professional. Ethical conduct is tied to professional practice.”³²

For Gotterbarn, then, the point of joining “ethical conduct” with “professional practice” is (in part at least) to stress that “professional practice” is “ethical conduct” (conduct made ethical by a moral “contract” between profession and society).

The commitment to something more than a code of ethics, if there was then such a commitment, may explain something else all three agreed to: adopting the *IEEE*’s “Standards-development model” (even though the *Joint Steering Committee* and its subcommittees belonged to the ACM as much as to the IEEE). For the IEEE, any Standard-developing committee (or other committee assigned a large project) becomes a “steering committee” that oversees the work of several “task forces”, one task force for each major division of the project. Each task force consists of several “working group leaders”, one leader (or occasionally more) for each subdivision of the project. Each working group leader oversees a specialized working group and reports back to the task force. The three agreed on a preliminary list of working groups: privacy; security; testing and reliability; professional competence; duties to society; intellectual property; and ethical tools.³³

The three also agreed to follow IEEE procedures for writing “Standards” (the capital “S” signaling a certain sort of standard). For the IEEE a standard (with a small “s”) is a general category consisting of “Standards” (with a capital “S”), “recommended practices” (marked by a “should” rather than the “shall” of a Standard), and “guides” (more than one practice can be recommended). Because Gotterbarn knew little about IEEE procedures, Melford undertook to prepare the required “Call for Participation” (CFP) the IEEE way. This is the first sign of a pattern that would endure for almost two years—with nearly disastrous consequences (as we shall see in subsequent chapters). Gotterbarn, the ACM’s co-chair, would defer to Melford (and Barbacci) on SEEPP procedure because the IEEE seemed to have (and actually did have) a good deal of experience writing standards, much more experience than the ACM, and writing a code of *ethics-and-professional-practice* seemed to be much more like writing other IEEE standards (whether Standards strictly so called, guidelines, or mere recommended practices) than like writing a mere code of ethics.

The three did not, however, rely entirely on the Call to bring in participants in the numbers and variety they wished. They had already recruited three members for the task force: Joyce Currie Little, identified in the minutes as an “educator”; Keith Miller, as the “ACM ethicist”; and Gerald Engel (newly returned to teaching after several years on loan to the National Science Foundation), as the “Grant Funding Advisor”.³⁴ Gotterbarn undertook to find a corporate ethics officer, and a lawyer with interests in malpractice, negligence, liability, and intellectual property. Melford undertook to contact some other “ethicists” (Patrick Sullivan and Michael McFarland); some experts in the IEEE’s Standards process (Leonard Tripp and Fletcher

Buckley); and the chair of the IEEE-CS ethics subcommittee that Melford had himself once chaired (Suzanne Weisband). While there was agreement that the task force or its working groups should include “technologists”, no one undertook to recruit any (perhaps because most of those already identified for other purposes fit that description).

Last, the three set a target date for completion of the work of the task force, November 1995. The task force would have twenty-two months, four months more than Anderson’s committee needed to write the ACM code (and win its approval). The three sketched a tentative schedule, beginning with the next meeting (“#1”) to be held on Saturday, April 30, 1994, and (after “meeting #4”) ending with “Submit (Nov. 95) report and draft to [IEEE-CS] BOG”.³⁵

The agenda of the planned next meeting (April 30) was ambitious. The meeting was to:

1. Revise Task Force schedule,
2. Partition/categorization of standards,
3. Funding grants,
4. Establish deadlines for WG [working group] draft standards,
5. Complete email/conferencing/public bulletin board,
6. Formulate draft PAR [Project Authorization Request] (for internal use only) to identify components for successful submission,
7. Prepare call for participation (CFP) in working groups,
8. Identify audience and time limit for CFP (e.g. 30 days),
9. Identify WG chairs (continues).³⁶

3.5 PAR in DC?

On April 24, Melford sent out an official agenda for the April 30 meeting; it summarized the January 29 meeting, refined the agenda items, and allotted time for each item (for example, “9:15-10:00, Define Scope of ethics TF [task force]; 10:00-10:15, Revise Task Force Schedule; 10:15-10:30, Library research: identify titles/documents, identify ‘who gets what’, and document repository”). The meeting was to end some time after 3:00 pm—after the task-force schedule had been “revised (again)”.

The timing and location of the meeting deserve comment. The meeting was scheduled in Washington, DC so as to come just after the annual meeting of the Computer Ethics Institute (CEI). The CEI was a good place to advertise SEEPP—as well as an important capital in the geography of computer ethics. CEI began in the mid-1980s as the Coalition for Computer Ethics, a local discussion group. Support came mostly from IBM, the Brookings Institution, and the Washington Theological Consortium, as did most participants. In 1992, it formalized itself as CEI, a project of Brookings. Though similar in name to SEI, CEI was (and remains) a quite different entity, “an advanced forum and resource for identifying, assessing, and responding to ethical issues associated with the advancement of information technologies in society”. CEI is supposed to stimulate awareness of ethical issues in technology by consultation, research, education, and public outreach, but *not* to set standards (though it did develop “The Ten Commandments of Computer Ethics”—a short code designed for all computer users—in the early 1990s). CEI’s primary focus seems to be organizing events bringing together people interested in public policy issues related to computers (both hardware and software).³⁷

Melford had been active in CEI from the late 1980s. That was how he knew Patrick Sullivan, its executive director since 1991. Sullivan was a philosopher (Ph.D. from the University of Kentucky, 1988, dissertation on Pierce's speculative rhetoric) who (like Gotterbarn) had developed an interest in computing. But (unlike Gotterbarn) Sullivan never became a programmer, much less a software engineer. Having followed his spouse to Washington in 1991, he combined some teaching (everything from introduction to philosophy at a small college nearby to the graduate bioethics continuing education program at Johns Hopkins). CEI was an ideal place for a philosopher to meet others with a sophisticated understanding of computer ethics.³⁸ Sullivan had met Gotterbarn at an earlier CEI conference. So, when Melford suggested that Gotterbarn be invited to participate in CEI's third annual Conference on Computer Ethics (*Building Ethical Culture in Cyberspace: Further Pursuit of a "Ten Commandments" for Computer Ethics*), Sullivan was happy to arrange a panel at which Gotterbarn (along with Melford) could discuss software engineering ethics. (The invitation meant that ETSU could cover Gotterbarn's travel costs—important because SEEPP still had no budget.)

SEEPP's meeting was held, as planned, at the Washington headquarters of the IEEE-CS—but that seems to be the only thing that went as planned. No minutes survive but, in a memo written a year later, Gotterbarn described the meeting in much less ambitious terms than the agenda had. A “get acquainted meeting”, it was a small affair, with only Gotterbarn, Melford, Steve Barber (a lawyer whom Gotterbarn had met a few weeks before)³⁹, Little, Sullivan, Ramon Barquin (president of CEI and a founder of the Coalition when he was at IBM), and Elliott Chikofsky (Steering Committee member who happened to be in town) physically present.⁴⁰ Buckley and Tripp participated by conference call. The discussion could not advance much beyond January 29 because the documents that such an advanced discussion presupposed (a draft of the Call for Participation, the PAR statement, and so on) were not ready. Gotterbarn recalls Melford asking Tripp detailed questions about how to prepare a PAR—with no one else present having any idea what they were talking about. Barber just remembers “a telephone conference with an IEEE process guru that seemed to go on forever with no real benefit. SEEPP members familiar with the process guru engaged in quite a bit of eye-rolling.”⁴¹

3.6 Report to Steering Committee

On May 28, 1994, at the Software Engineering Institute, the Joint Steering Committee held its next official meeting (starting at 10:00 and ending at 5:00, with an hour for lunch). All Committee members were present except Buckley, Melford, and Shaw.⁴² Each task force reported, beginning with the body of knowledge (Douglas) and ending with the curriculum (Carver and Werth). What each reported was a sketch of the work ahead. “Discussion” followed. The meeting's minutes use one page single-spaced to summarize the three reports but three-and-a-half pages to summarize the “Discussion”. Of those three-and-a-half pages, about two thirds concern the body of knowledge, with the remainder divided evenly between curriculum and ethics. (The schedule of the meeting reveals a similar emphasis, with the body of knowledge receiving seventy-five minutes to sixty for ethics and forty-five for curriculum.) Clearly, for the Steering Committee, defining the body of knowledge was the most interesting, or at least the most troubling, part of their project. But our concern is SEEPP.

Gotterbarn spoke for his task force. All members of a profession have obligations because they belong to that profession. Reflecting on the obligations of other professions, the

task force identified three “orthogonal axes” by which other professions organize their special obligations. One is the *groups* to which they have special responsibilities (the individual, the organization, the society). A second axes is the *standards* from which the obligations derive (justice, equity, privacy, autonomy, beneficence, avoidance of harm, fidelity, integrity). The third axis is defined by *problems* (privacy, security, testing and reliability, competence, duties to society, intellectual property, and ethical tools).⁴³ The point, apparently, was that though SEEPP seemed so far to be working as if the third axis (the problems) were the only possible axis, it was aware of the other two. The other two were certainly potentially important. For example, the old IEEE code of ethics (1987) was organized by group, while the new one (1990) was organized by standard.

Gotterbarn also indicated that SEEPP intended to follow the IEEE Standards process the scribe “correcting” Gotterbarn’s description with a bracketed “actually, ANSI” Standards process). While the minutes do not suggest that anyone on the Steering Committee had any doubts about using the IEEE Standards process, Gotterbarn recalls a few familiar with it (Druffel, Chikofsky, and perhaps someone else) privately recommending against following the whole elaborate process, because the process usually took “too long” (that is, longer than anyone wanted the work of the Steering Committee to take). Having had no experience with the IEEE process, Gotterbarn had no idea that “too long” meant “five to seven years” (the normal time for developing a standard). He was still learning his way around IEEE-CS.

The recorded discussion of Gotterbarn’s report concerned the “target” of the standards SEEPP would develop. Most of that discussion (as the minutes report it) took the form of a four-layer chart. At the top is “User Programming (100 million people in US)”. The next layer consists of categories of people: “Applications Generators (0.2 M)”, “Application Composition (2 M)”, and “Systems Integrators (2 million)”. Below these are “Infrastructure (1 million)”; and, at the bottom, “Academics Foundation Infrastructure (0.01 million)”. The discussion concluded that the task force should target “the middle layer”. This leaves the impression that the target is to be everyone between User Programming and Academics, more than five million people in the US alone, a group at least five times larger than any published estimate of the number of “software engineers”. So, perhaps, the Infrastructure group was what was actually intended. (Gotterbarn believes that this chart was Boehm’s contribution: “Boehm was always ‘chunking’ [that is, grouping ideas or making distinctions].”⁴⁴

The only “action” the Steering Committee took relevant to SEEPP was to assign Barbacci and Zweben responsibility for a press release from the “TF on Ethics and Professional Practices through CS/ACM... at House briefing on June 21”.⁴⁵ For those already cynical about the IEEE-CS and ACM undertaking to make software engineering a profession, this briefing will seem an important event. Participants not in the Washington area (including both Gotterbarn and Melford) had expenses paid by IEEE-CS or ACM. They were brought to a meeting room in the House’s Rayburn Office Building and seated in a row facing a few reporters.⁴⁶ During a presentation that lasted only a few minutes, the speaker said something like “We are working on ethical issues in computing” and pointed to Gotterbarn and Melford. They had no opportunity to speak or answer questions. Gotterbarn had never seen the speaker before—and never saw him again. Soon their part of the briefing was over and the co-chairs were on their way out of town, Gotterbarn having no idea who had arranged the event or why.⁴⁷

3.7 Defining the working groups

On June 11 (ten days before the briefing), the task force held its second scheduled meeting. The host was Michael McFarland, then a professor of computer science at Boston College.⁴⁸ The location was a local monastery. Of the eight members of the task force, six were physically present—with a seventh (Weisband) present by conference call. Only Barber could not attend even by speaker phone. The first half of the meeting consisted of a sorting out of research tasks: There was a need for an online library to hold codes of ethics (IEEE, ACM, British Computer Society, Australian Computer Society, and so on, including several European codes), bibliographies, course syllabi on computer ethics, conference programs, case studies, and so on. McFarland would find a piece on casuistry to include. A graduate student “from somewhere” would manage the library. We never again hear of this library.

There followed a discussion of how to circulate the (yet to be prepared) Call for Participation. The first venues discussed were all list servers or electronic bulletin boards, with paper publications (*Communications of the ACM*, *SIGCAS*, and so on) considered next. Melford (again) undertook to prepare a “general boilerplate” Call and circulate it to the working group chairs.

The eight working groups were tentatively set up, that is, a chair (drawn from the task force) was named for each. McFarland took on *Institutional Support*; Gotterbarn, *Professional Competence*; Weisband and Barber, *Intellectual Property*; Little, *Professional Relations*; Sullivan, *Privacy*; Miller, *Reliability and Safety*; Melford, *Security*; and Weisband, *Social Justice*.⁴⁹

We might make three observations concerning these working groups. The first is that their titles (or “problems”) have changed a bit since May. “Institutional support”, “professional relations”, and “social justice” have been added; “duties to society” and “ethical tools” have disappeared. Perhaps “social justice” is a mere synonym for “duties to society”, but “ethical tools” does not seem shorthand for “institutional support” or “professional relations” (or any obvious combination of them).

The second observation is that understanding the architecture of this list of “problems” (or the one it replaces) is difficult. The list does not seem exhaustive. Some common issues of software engineering ethics, such as conflict of interest or truth in advocating a design option, do not seem to fit snugly under any of the eight categories. The list also seems not to consist of mutually exclusive categories. Some issues, such as protecting information or protecting end-users from harm, seem to fit under two or more of the eight categories (say, safety and social justice). The looseness of the architecture seems to have been noticed. Half the meeting’s minutes consist of a “discussion of distinguishing content”. So, for example, “Professional Competence” is analyzed as “avoid generalization of expertise, keep current, keep staff current, truth about skill, what should be produced, appropriate knowledge base”. “Privacy” is distinguished from “Security” because privacy is “Nightline shows, Monitoring (workplace, everywhere), data aggregation, data valence—digital persona, misuse of data” while security is “illegal stuff, data integrity, trap doors, system administrator responsibility, malicious programming, encryption”.

Clearly, the question of architecture remained. This suggests that we might better understand the division of labor among working groups by looking at who was assigned to chair the group. Since no one now recalls how this list came about, we must try to reconstruct what might have happened. The only obvious hypothesis is that there is a working group where there

is a potential chair already interested in the problem. We can find two sorts of evidence for this hypothesis (beyond the lack of obvious architecture).

The first consists of what we know about the interests of the working-group chairs. For some, the evidence is pretty clear. Melford made his living making computer systems secure. His interest in security went back to his undergraduate days.⁵⁰ Gotterbarn (as he made clear to Barbacci) thought of ethics as a way of helping to assure professional competence. McFarland had published two papers on computer ethics, each with a final section on structuring the institutions in which computer professionals work to help them do the right thing.⁵¹ Long active in the Institute for Certification of Computer Professionals (ICCP), Little was interested in relations among professionals (which she understood as primarily professional organization).⁵² Miller had published a half dozen articles (and had given a larger number of papers published in transactions) on testing and reliability in the four years just before joining SEEPP.⁵³ Sullivan's interest in privacy had been developing for several years and would lead him, only three years later, to leave CEI to join Price Waterhouse's new consulting group concerned with compliance with U.S. and international privacy and data protection regulations.⁵⁴ Weisband would have brought a concern for social justice with her from the Committee on Public Policy.⁵⁵

Barber appears to be the chief exception. He was recruited because he specialized in intellectual property; yet he is not chair of the working group on intellectual property; Suzanne Weisband is. Weisband was first named as a possible participant in the organizing meeting on January 29 without any subject being mentioned. But she had recently published two (short) papers on software piracy.⁵⁶ Perhaps she reminded Melford of that when he contacted her in February to invite her to SEEPP's April meeting. She is listed ahead of Barber under Intellectual Property, out of alphabetical order, the normal sign of serving as chair. Neither Barber's interest nor legal credential was enough to get him his own chair. Apparently, Weisband got the chair because she outranked him (or because he was absent); Barber had to settle for vice chair. Thus Barber is one of those exceptions that, while not necessarily proving the rule, at least does not disprove it.

The working group chairs are, admittedly, knowledgeable about the problems assigned them as well as interested. But their knowledge does not explain the list. Knowledge (or competence) would explain why they were appointed as chairs once the list had been drawn up but something else must explain the list. On the one hand, knowledge did not guarantee that one would be the chair. Barber doubtless knew more about intellectual property than Weisband and she had another chairmanship (Social Justice). She nonetheless took the chair of Intellectual Property, leaving Barber to be vice chair. On the other hand, the working-group chairs doubtless knew much about many problems not on the list, such as continuing education or enforcement of standards. They must also have thought many other problems important enough for SEEPP to consider; they had, after all, earlier produced a somewhat different list of problems for the working groups. But if no one expressed an interest in some of the original problems, those problems would disappear from the list.

The second sort of evidence for the hypothesis that the list of groups mirrors the interests of participants rather than any plan for a code of ethics is in the Call for Participation. The Call (in all published versions) speaks of "working groups...being formed to address specific areas of *concern*" (my italics). Though eight are then identified, the Call adds, "Should additional subject areas be identified, additional working groups may be established."

3.8 The Call goes out

On August 24, Melford faxed Gotterbarn a draft of the Call for Participation. A week later, Gotterbarn sent all working group leaders the draft, asking for comments. Having received almost no comments by the end of October, Gotterbarn made a few changes and sent out the revised draft for “final” comment by November 5, 1994. As soon as the deadline for comment had passed, he sent out the announcement to all the publications previously identified as appropriate venues for the Call. The Call was a substantial document, four pages single-spaced (with a two-page application form appended). While the official Call did not go out until November, Melford seems to have begun sending out (short) unofficial versions earlier (at least as early as June)—without telling Gotterbarn.⁵⁷

The Call (November 5) begins “Dear Computer Professional”. It had six main divisions (as well as an untitled introductory paragraph explaining the larger IEEE-CS/ACM undertaking). The main divisions are: Scope, Purpose, Organization, Participation, Internet Access, Completion, and Thank You. Each of these main divisions is numbered, beginning with “1.0” for Scope and ending with “6.0” for Thank You. Most divisions are quite short (a single sentence for all but Scope and Organization). But Organization, with eight subdivisions, covers two-and-half pages.

The Scope 1.0 states that the task force would “document generally accepted principles for identifying and resolving ethical conflicts” in software engineering—suggesting a (descriptive) survey rather than a (prescriptive) codification. The Scope then provides a list of stakeholders (persons or groups whom the software engineer might have special responsibilities or obligations to: “peers and laypersons, employer, customer(s), the profession, and society/humanity”). But the Scope also includes in the task force’s work “consideration... [of] the obligations and responsibilities of these various entities towards the Software Engineer” (something the Purpose statement immediately following omits). This reference to obligations of those who are not software engineers suggests a document (like the American Medical Association’s 1847 code of ethics) that states *reciprocal* obligations between an occupation and other stakeholders (“rights and responsibilities”), not—like a typical professional code—a simple statement of the profession’s obligations. We never again hear of the obligations of others to software engineers.

Scope 1.1 adds that the task force recognizes that the profession “transcends national boundaries”. Any standards the task force developed should “be as reflective of the global computing community’s wisdom as can be reasonably achieved.” The standard will state what is generally accepted or, where no one standard is generally accepted, “various recommended practices and guidelines”.

Organization 3.0 describes the eight working groups in more detail than the June minutes, but in much the same terms—except that Barber is now explicitly designated as vice chair of the working group on intellectual property (with Weisband as chair).

Participation, the first subsection under Organization, contains several surprises. 3.1 declares participation “open to all individuals who are directly or materially affected or interested in” any issue within the Scope of the working group. “Members of the international computing community” are “particularly encouraged to participate”. Essentially, *anyone* who

wants to participate can, presumably even a gaucho on the Argentine plains or an Imam in the holy city of Qom. Participation 3.1.1 adds that individuals may participate in as many working groups as they desire. 3.1.2 allows organizations to send representatives but warns that no organization will have a “veto”. After all these inviting words, 3.1.3—all in caps—comes as a surprise: “PLEASE COMPLETE THE APPLICATION AT THE END OF THIS DOCUMENT AND E-MAIL IT TO SEEPP—CFP@COMPUTER.ORG, BEFORE 5-NOVEMBER-1994. We are organizing the working groups on 12 November 1994.” The November 12 deadline—at most, a week from the date on which the Call was issued—is reasonable if, but only if, the expectation was that great numbers of potential participants were ready to apply and most of them would read the Call soon after it was sent out. If, however, filling out the form was demanding enough to be put off for a time for other things, or if many who might read the message would wait till it had been printed and distributed, or if the pool of potential participants was in fact small, any deadline, but especially one so tight, seems to be a mistake. It is a good sign, then, that when Gotterbarn and Melford met again, along with Sullivan, on November 11, at a computer ethics conference in Washington, DC, they already had two dozen or so applications to look at.

Later versions of the Call, such as that printed in the December 1994 issue of *Computers and Society*, never again contained a deadline for application. Earlier versions of the Call—for example, the preliminary October version that Gotterbarn had sent out to some lists in October⁵⁸—also contained no deadline. The November 5 deadline thus seems to have been a (temporary) attempt to speed the process rather than a device to keep the number of participants down—and it seems to have worked.

3.9 A good report

A few days after the November meeting, Gotterbarn sent the list of those responding to the Call to all the working group leaders, asking them to identify anyone interested in their working group, organize mailing lists for their group, and get to work. By then, the applicant pool was over thirty (with many indicating they wanted to participate in two or more of the working groups).

So, when, on November 18, the Steering Committee had to report to the IEEE-CS Board of Governors on its activities for the year, it could describe SEEPP’s achievements with at least as much pride as those of the other two task forces:

Meetings

- April 30, 1994, CS Headquarters
- June 11, 1994, Boston College
- November 11, 1994, Ethics in the Computer Age Conference

Working groups

- Institutional support
- Intellectual property
- Professional competence
- Professional relationships
- Privacy
- Reliability and safety

- Security
 - Social justice
- Call for participation in wide circulation

These achievements were apparently substantial enough that, in December, Barbacci undertook to cover the task force's overhead (a few conference calls and some travel).⁵⁹ The payment of these expenses seems significant only because the Call makes clear (Finances 3.1.3) that most expenses will not be covered: "Participation is voluntary. Each individual is responsible for expenses incurred to support his/her participation. Many institutions provide support for their volunteers."

NOTES

¹ Dennis Frailey has BS in mathematics from Notre Dame University (1960) and an MS and PhD in computer science from Purdue (1971). He claims to have worked in software engineering before there was such a field (that is, since 1962). He worked for Texas Instruments and Ratheon while also serving as an adjunct professor at Southern Methodist University. See <http://engr.smu.edu/~frailey/bio.html> (August 28, 2004). Frailey became vice chair when Zweben, having become AMC President, became a mere *ex officio* member.

² While Hoelzeman is listed as a member *ex officio*, that designation seems to be a mistake. The "ex officio" are task force chairs. Hoelzeman is not a task-force chair (and seems to hold no other office that would entitle him to that designation—though he was then a member of the IEEE-CS Board of Governors). He can only be Schneidewind's replacement.

³ John Werth has a BS and MS in Mathematics, Emory University (1962 and 1963) and a PhD in Mathematics from University of Washington (1968). He was at this time Senior Lecturer and Research Scientist, Computer Science Department, University of Texas, Austin. <http://www.cs.utexas.edu/users/jwerth> (August 28, 2004).

⁴ Laurie H. Werth (John Worth's spouse), was an early SEEPP volunteer. She too taught computer science at the UT-Austin. Gotterbarn\SEEP1994-95, sec. 11.

⁵ See Chapter 2.3.

⁶ http://sunset.usc.edu/Research_Group/barry.html (August 28, 2004). I have corrected some dates of publication.

⁷ Feldman declined to be interviewed for this book, claiming to have had nothing to do with SEEPP, even though his own biography lists him as a member of the Joint ACM-IEEE task force. See <http://www.research.ibm.com/iac/advisory-feldman.html> (August 28, 2004). Chapter 8.6 describes Feldman's part in writing a 1997 disclaimer that SEEPP was to put in published versions of the code stating that the publication of the Code draft (Version 3) did not indicate any approval of the Steering Committee. Perhaps this was the extent of Feldman's participation and, from his perspective, not a significant contribution to the code.

⁸ I have not been able to find a written source for these estimates.

⁹ An affiliate member of IEEE-CS has vote in the CS but not in its parent IEEE.

¹⁰ <http://www.ieee.org/membership> (August 28, 2004). Since 2001, ABET and CSAB have worked together on accreditation, dispatching joint committees.

¹¹ <http://www.acm.org/membership> (August 28, 2004).

¹² That is, he did not receive a degree from an ABET-accredited engineering program and probably could not be licensed as a P.E. (since he lacks an appropriate degree). Or, as Melford himself put it, in answer to the question, “Are you an engineer?” (Interview of Melford, November 1, 2002): “Not officially. I’m not licensed and I don’t have a degree. But most people would consider me to be an engineer. I think most of my colleagues view me as such. I certainly identify with that side of my field, probably more than anything else other than general business. I am an engineer in the sense that I determine how technology can be applied in a cohesive and complimentary manner to solve problems.” See also his answer to question 1, about his education (endnote 15 below).

¹³ Interview of Melford, November 1, 2002.

¹⁴ “I left school to earn a living. I was studying economics. I have some formal education in economics and some in programming. I am fully self-educated in systems management as well as the business side of computing operations. I am fully self-educated in networking, the technical aspects of networking, as well as the ethics. One of my incentives for leaving school was that I wasn’t able to get what I was after. At the time, there was basically nothing available on computing security. I originally went in to study electrical engineering, but found that that wasn’t quite my forte. I came into computing in a backward sort of way. I read some issues of *Scientific American* in the late ‘70s or early ‘80s. I believe it was the September 1980 issue that was devoted to automation in the workplace, and the following issue in October was dedicated to software computing. Those two articles were extremely influential on me. They made me think about automating the business workplace. During that time, I had taken several courses in programming and management of information systems (MIS). I found that I had a knack for it. I also took programming classes that were oriented toward business majors, towards accounting students. That was at Cal State Northridge. When I left school I decided that I wanted to continue my education on my own. So I went and bought a PDD11. Yes, that was the computer Digital Equipment Corporation then made, the old deck PDD11, forerunner of the Vax. I mounted an early version of UNIX on it, and started to go from there. That led into some consultant work. I started my own consulting practice in 1984. I did consulting for 15-16 years.” Interview of Melford, November 1, 2002.

¹⁵ Gotterbarn’s vitae also lists (among philosophy publications during this period): “A Model for Software Engineering Ethics” in *Lecture Notes in Computer Science: Software Engineering Education* (Springer Verlag, 1989). Amid this philosophy, are Gotterbarn’s first

publications in computer science: “An Optical Bus Architecture”, Technical Report F86-2, *Allegany College Technical Report Series* (1986); and a contribution to pedagogy, *A Study Guide for Hexadecimal Arithmetic* (Harrisburg Area Community College, 1980).

¹⁶ Gotterbarn could be a “faculty tutor” because he was also teaching philosophy at HACC (Introduction to Philosophy and Logic).

¹⁷ Thomas J. Deloughry, “Failure of Colleges to Teach Computer Ethics is Called a Oversight with Potentially Catastrophic Consequences”, *Chronicle of Higher Education*, February 24, 1988.

¹⁸ Gotterbarn later wrote the first encyclopedia article on software engineering ethics for the *Encyclopedia of Software Engineering* (1994), p. 1197-1200.

¹⁹ The ACM pays the airfare of a Lecturer for a visit to an ACM professional or student chapter. The local organization covers food, hotel, and other local expenses.

²⁰ Anderson is (and was then) a sociologist at University of Minnesota (Ph.D., Stanford, 1970). For an explanation of the code—and a brief history of writing a code much less messy than the one described here—, see Ronald E. Anderson, “The ACM Code of Ethics: History, Process, and Implications”, in Chuck Huff and Thomas Finhold, editors, *Social Issues in Computing: Putting Computing in its Place* (McGraw-Hill, Inc.: New York, 1994), pp. 48-62.

²¹ The most important of these conferences were: the Computer Science Conference and the National Educational Computer Conference.

²² After Anderson, the ACM code lists as authors (with the parenthetical affiliations being my addition): Gerald Engel (University of Connecticut-Stamford, but then on loan to NSF), Donald Gotterbarn (ETSU), Grace C. Hertlein (California State-Chico), Alex Hoffman (Texas Christian University), Bruce Jawer (IBM), Deborah G. Johnson (Rensselaer Polytechnic Institute), Doris K. Lidtke (Towson State), Joyce Currie Little (Towson State), Dianne Martin (George Washington), Donn B. Parker (Standards Research Institute, but with an early book on computer ethics), Judith A. Perrolle (Northeastern University), Richard S. Rosenberg (University of British Columbia). Note that all but two (that is, 4/5’s—not counting Anderson) are academics. Anderson, “ACM Code History”, does not explain how these few names were chosen from the larger number of those who participated in some way or another in writing the ACM code.

²³ Among others whom Gotterbarn first met at the conference and would later recruit for SEEPP was a philosopher at the RCCS (John Fodor, who had only recently received a Ph.D. from Washington University-St. Louis).

²⁴ Donald Gotterbarn, “Two Computer-Related Codes”, *Perspectives on the Professions* 19 (Fall 1999): 5-6

²⁵ This paper eventually appeared in the *Communications of the ACM* (February 1993): 98-108. Judith Perrolle also listed as authors. This paper and presentation were Johnson's only part in "writing" the ACM code of ethics. Apparently, one can become an author by lending one's authority to a text (authorizing it) as well as by helping to write it (authoring it).

²⁶ Gotterbarn, "Two Computer-Related Codes".

²⁷ X (memo's name) gives the date as "November 10". That seems to be a typo.

²⁸ Barbacci, email, September 12, 2005.

²⁹ Undated Gotterbarn memo beginning "Mario, I enjoyed meeting". (paper only).

³⁰ Gotterbarn's emailed comments on March 24, 2003 draft of this chapter, p. 9.

³¹ Gotterbarn, email, April 24, 2003.

³² Undated Gotterbarn memo beginning "Mario, I enjoyed meeting" (paper only).

³³ Anderson had never divided work on the ACM code into parts. His group had always treated the code as a single indivisible undertaking. Hence, the decision to divide SEEPP into working groups does not seem likely to have originated with Gotterbarn.

³⁴ Gotterbarn knew Engel from ACM (see below). Barbacci and Melford also knew him—from IEEE-CS.

³⁵ Gotterbarn\SEEP1994-95\FIRST meeting 1-94.

³⁶ Gotterbarn\SEEPP1994-95\SEEP2A.

³⁷ http://www.brook.edu/its/cei/cei_hp.htm (August 28, 2004).

³⁹ This was at the fourth "Conference on Computers, Freedom, and Privacy", held at the John Marshall Law School, Chicago, March 23-26, 1995. Gotterbarn, who had organized the panel ("The Licensing of Computer Professionals"), is identified in the panel's abstract as "chair of the SIGCAS Task Force on Licensing and Certification of Computing Professions". Barber was on the panel (more or less) by accident. Gotterbarn had initially asked Lance Rose, a New York City attorney specializing in software and online intellectual property. Rose had previously spoken on licensing to a software consultants' group in New Jersey. Since Rose was doing other things at the conference and felt he did not have much more to say on the panel's subject, he suggested one of his associates, Barber, instead. Barber, though new to lawyering (J.D., Yeshiva University 1993), seemed a good choice. He had a B.S. in Computer Science and Engineering (MIT, 1987). From 1984-1990, he had worked as an in-house developer for a vendor of packaged commercial software. He had worked on IBM protocol emulation packages for PCs and Unix-based systems. He was exposed to and performed tasks related to almost all phases of

the software development lifecycle—module design, implementation, testing, in-house and field maintenance, and project management. He also co-wrote and edited a technical book (Lewis, Barber, and Siegel, *Programming in Java IDL*, Wiley 1987). Gotterbarn, impressed by what he heard (arguments *against* professionalization), invited Barber to dinner, laid out his plans for SEEPP, and invited Barber to join. Interview of Barber, November 14, 2002.

⁴⁰ Gotterbarn\History of SE Code\History expanded, under 4/95 meeting: “partial list of attendees. Robert Melford, Don Gotterbarn, Steve Barber, Joyce Currie Little, Patrick Sullivan, and Fletcher Buckley and Suzie Weisband on conference call. [where did I find out about Chikofsky and McFarland?]”

⁴¹ Interview of Barber, November 14, 2002.

⁴² There was also a “guest” present, Sam Redwine, a software engineer, at that time Assistant to the President and CEO of the Software Productivity Consortium, Herndon, VA. He is now an Adjunct Professor, Information and Software Engineering, George Mason University. Highly respected within IEEE-CS, Redwine may have been present simply because he was in town and knew a lot about IEEE procedures.

⁴³ While the minutes give the impression that these three axes were the work of SEEPP, Gotterbarn claims that Barry Boehm suggested them during discussion. (Emailed comments on draft of chapter, Gotterbarn, March 24, 2003.) Gotterbarn’s claim may provide a good example of the unreliability of memory. Gotterbarn’s Archive contains an undated document (SEEP1994-96, STEER.M1A), apparently the written report Gotterbarn and Melford submitted to the Steering Committee in advance of the May meeting (the computer file bears the date of May 27, 1994, one day *before* the meeting). This document explicitly discusses the “three axes”. So, if the file date is right, then the axes are (one of) Gotterbarn’s contributions, not Boehm’s.

⁴⁴ Gotterbarn email to Davis (March 24, 2003).

⁴⁵ As Gotterbarn (March 24, 2003) recalls events, Zweben was to help draft the release because he had raised several objections to the version Barbacci had presented at the Steering Committee meeting. The minutes give no indication of this. Gotterbarn also recalls Barbacci saying that he was still seeking funding to help support SEEPP’s work. The minutes make no mention of that either.

⁴⁶ Gotterbarn, who already had an airline ticket to Boston to attend a conference, only billed ACM for \$60 to break his journey in DC and \$22 for cab fare to and from Washington National. Gotterbarn Archive\SEEP1994-96\ACM expenses.

⁴⁷ Gotterbarn (March 24, 2003). Gotterbarn thinks Melford knew more about the event than he did. Gotterbarn (June 5, 2003, comments on Ch. 2).

⁴⁸ McFarland has B.A. in Physics from Cornell University (1969); M.A. and Ph.D. in Electrical Engineering from CMU (1975). He joined the Jesuits in 1975, and was ordained in 1984. He worked as a consultant at AT&T's Bell Labs in Murray Hill, New Jersey, for two years (before going to Boston College). He is now President of the College of the Holy Cross, Worcester, Massachusetts. His best-known paper in ethics then was: "The Public Health, Safety, and Welfare: An Analysis of the Social Responsibility of Engineers", *IEEE Technology and Society Magazine* 5 (December 1986): 18-26, already reprinted in Deborah G. Johnson, *Ethical Issues in Engineering* (Prentice-Hall: Englewood Cliffs, New Jersey, 1991), pp, 159-174.

⁴⁹ This assignment does not appear in the official minutes Melford prepared (Gotterbarn/94-95 misc/Boston 6-94), only in Gotterbarn's 1999 reconstruction of the preceding six years of work (addressed to "Leonard" Tripp). Melford's minutes merely conclude with the words "Preliminary assignment of working group chairs" (suggesting that matters may have been a bit less settled). Melford's minutes also omit Institutional Support altogether, defining Professional Relations to include it—"P-P (support for whistle blowers). P-org, P-individual, P-society, P-electronic community, obligation to foster responsible behavior in the non-professional". It is, however, easy to see how Institutional Support might have been carved out of this collection of topics.

⁵⁰ Interview of Melford, November 1, 2002.

⁵¹ Michael C. McFarland, "Urgency of ethical standards intensifies in computer community", *Computer* (March 1990): 77-81; and Michael C. McFarland, "Ethics and the safety of computer systems", *Computer* (February 1991): 72-75. These two papers appear in the "Standards" section of the magazine, with Fletcher J. Buckley listed as "Editor"; one of Melford's papers is cited in the second of McFarland's. (McFarland was one of Melford's recruits.)

⁵² Interview of Little, June 11, 2003.

⁵³ See his curriculum vita: <http://www.uis.edu/~miller/keith.html> (June 28, 2004)..

⁵⁴ Interview of Sullivan, November 10, 2002.

⁵⁵ Suzanne P. Weisband, Associate Professor of Management Information Systems, School of Management, University of Arizona. B.A. in Organizational Psychology (San Diego State University, 1979); Ph.D. in Social and Decision Sciences and Policy Analysis (Carnegie Mellon University, 1989). She is not a programmer but an expert on the "human factors" side of computing. <http://www.bpa.arizona.edu/~mis/faculty/sweisband.shtml> (August 28, 2004).

⁵⁶ S.P. Weisband and S.E. Goodman, "International Software Piracy", *IEEE Computer* (Nov. 1992): 87-90; and S.P. Weisband and S.E. Goodman, "Subduing Software Piracy", *Technology Review* (Sept./Oct. 1993): 30-33.

⁵⁷ Gotterbarn \by year\1993-4\Establishing the CFP (November 5 version). In an *October* 11, 1994 email, Jon Fineman (Gotterbarn Archive\SEEP1994-96\HELPOFR) says it was out in *IEEE Computer*: “Sorry to hear the world is giving you a hard time. I am confused though, what I was responding to looked like a call for participation that was advertised in the IEEE Computer magazine. That’s what generated the note, since I did not want to miss the boat. Do you need any help getting things organized.” Burnstein’s (paper) file includes an even earlier version of the Call, a “Press Release from Melford” dated June 14, 1994. Gotterbarn was doing something similar. See Interview with Rogerson, February 24, 2003 (quoted below).

⁵⁸ See, for example, Interview with Rogerson, February 24, 2003: “On 19 October 1994, I received a broadcast email [of the Call for Participants] from Don Gotterbarn (sent through the Computer Ethics Institute list, to which I belonged). I’ll give you a copy. I sent in a filled out form the next day.” I never got the promised copy.

⁵⁹ The only evidence (in the archives) that Barbacci in fact did obtain a budget for SEEPP is a request from Gotterbarn in a letter (February 21, 1996) addressed to Curtis Harris at ACM requesting reimbursement for expenses of a meeting attendance at which had been authorized by Dennis Frailey. Gotterbarn\SEEPP1006-97\SEPBIL.

Copyright © 2009

This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.