Chapter 12: End Game, Version 5.2, 1999-2000

> "It isn't that things will necessarily go wrong (Murphy's Law), but rather that they will take so much more time and effort than you think if they are not to go wrong."
> —Wolf's Law

12.1 The Joint Steering Committee's happy end

On November 20, 1998, the Joint Steering Committee presented its "annual report" to the IEEE-CS Board of Governors. Prepared by "Dennis Frailey, co-chair and Leonard Tripp, chair", the report began by describing five "highlights" of an evaluation of the Committee's operations "for the period 1993 to 1997". (The evaluation had been undertaken soon after Tripp took over six months before.)[1] These "highlights" (under the heading "Committee Reorganization") seem to represent the thinking of the Committee, or at least of Tripp and Frailey, early in 1998. The first "highlight" was that the IEEE-CS and ACM should "continue the work begun by the steering committee." What the Committee was doing was worthwhile; it only needed to be done in a better way. Second, the Body of Knowledge task force would require "professional staff support" to finish its work. The work had proved too much for a task force consisting entirely of volunteers (and specialists they could hire under grants they were able to obtain). The third evaluation should be familiar (from 11.2): the "current draft of the Code of Ethics" (Version 4) is ready "for review in a formal consensus process with a balanced representation in the review team". Fourth, there was no reason for the Curriculum task force to wait for the Body of Knowledge task force to complete its work. The preparation of "curriculum criteria is ready to commence". The last "valuation" was a recommendation: that the ACM and IEEE-CS jointly sponsor a "permanent entity" to carry on the work of the Steering Committee (and, by implication, that the Steering Committee be disbanded).

This section of the report ends with the announcement that a "charter for the new committee" was prepared in 1998, approved by the ACM Executive Council and the IEEE-CS Board of Governors, and put into operation. The first "Executive Committee" of this new entity—the "Software Engineering Coordinating Committee" (SWECC)—had six members (listed in alphabetical order). Three of the names are new: Mark Ardis (Bell Labs), Linda Northrop (Software Engineering Institute), and Karl Reed (Department of Computer Science and Computer Engineering, La Trobe University, Australia). The other three are familiar: Carver, Failey, and Tripp. Of the six, only one (Frailey) is from the original Joint Steering Committee (Carver having been a member only *ex officio* for the Committee's first two years). According to a working document (September 29, 1998), Frailey was to be "Vice-Chair", representing the "ACM EC"; Ardis, to be the ACM member representing the "technical community" (and endorsed by SIGSOFT); and Northrop, to be the ACM member representing the "education community" (endorsed by the "ACM Ed Board Chair").[2] Those labels suggest that the other three are the IEEE-CS members: Carver (IEEE-CS President) representing the IEEE-CS Board of Governors; Reed (a professor), representing the "educational community"; and Tripp (at Boeing), "the technical community". In fact, when they were listed in the minutes of SWECC's first meeting, Tripp was only identified as "Chair". Carver (rather than Reed) was identified as

"CS Education" (that is, the representative of the "education community") and Reed (rather than Tripp) as "CS TCSE" (the representative of the "technical community").[3]

In its fifty-fifty split between ACM and IEEE-CS appointees, SWECC's executive committee is plainly the Steering Committee's successor. It is also the Steering Committee's successor in having under it "project committees" that, at first glance, seem to be "task forces" under a new name.[4] In several other respects, however, SWECC is a different entity. First, it was to have only six members, roughly half the number of the old (ten-member) Joint Steering Committee (excluding *ex officios*), but exactly the number the IEEE-CS Steering Committee had before the ACM joined late in 1993. Apparently, ten had proved too large for the purpose. Second, though SWECC may have as many project committees as required, it could create them only with "the consent of the Executive Committee of the ACM and the Executive Committee of the [IEEE-]CS". Each project would have "an approved duration".[5] The project would go out of existence after that date—unless the ACM and IEEE-CS authorized an extension. Projects were to find it harder than task forces to go on year after year. Third, the chair was to have a two-year term and rotate between the two societies. The first chair was to be from the IEEE-CS (and to be appointed by its Board of Governors); the next, appointed by the ACM Executive Council (from among SWECC members); the third, appointed in the same way by the Board; and so on. The ACM and IEEE-CS were to be exactly equal in SWECC. Indeed, this equality was to extend to the projects. When SWECC "formed a project", it was to create a "committee for its oversight with equal membership from the ACM and the CS." SWECC could, it seems, avoid the disastrous double-chairing that had nearly sunk SEEPP, but it could (it may seem) never again allow an executive committee like that Gotterbarn set up in January 1997. The next Gotterbarn would have to choose an executive committee with an equal number of ACM and IEEE-CS members; the next Patricia Douglas would have to find an ACM co-chair right away; and so on.[6]

We should not be too quick to treat this prospective elimination of the "next Gotterbarn" as a condemnation of the actual Gotterbarn—or as nearly as important a constraint as it sounds. In fact, the overlap between ACM and IEEE-CS allowed Gotterbarn to keep his executive committee just as it was when (as we shall see in12.3) SEEPP became "SEPEP" (one of SWECC's projects). Gotterbarn could be a member for both "ACM and IEEE-CS", Miller, for IEEE-CS alone (though he was also a member of ACM); and Rogerson, a member of ACM (and the British Computer Society) but not of IEEE-CS, could counter-balance Miller. That preserved equality between the two societies—if only technically.[7] The ease with which the executive committee of SEEPP's successor satisfied the new requirement is a reminder that no one seems to have objected to how Gotterbarn organized SEEPP. Indeed, his handling of SEEPP earned him a "retroactive promotion". Early in January 1999, Tripp asked Gotterbarn to provide a timeline for SEEPP. Gotterbarn did as asked, explicitly mentioning his co-chair, Robert Melford, at appropriate moments.[8] Yet, when the Steering Committee published its "History" (in March 1999), there was no mention of Melford. The history lists Gotterbarn as "chair" for the entire period, 1994-98 (as if there had been no reorganization at the end of 1996). What makes the lack of any mention of Gotterbarn's co-char seem significant is that the History lists all the other task force chairs (all as "co-chairs"), even those who (like Carver and John Werth) had been out of office for as long as Gotterbarn's co-chair.[9]

SWECC differed from the Joint Steering Committee in another way worth mention. The purpose of the Steering Committee had been to *establish* software engineering as a profession. That purpose at least suggested that software engineering was not yet a profession. In fact, when the ACM joined the IEEE-CS Steering Committee, it had expressed doubts about whether having a profession of software engineering was even a good idea (2.5), apparently embodying those doubts in the appointment of Mary Shaw as one of its representatives. SWECC's charter suggests that the general understanding of software engineering may have changed between 1993 and 1998. SWECC's purpose, according to the charter, was "to foster and maintain software engineering as a professional computing discipline." While "foster" still expresses doubt that software engineering is *fully* a "professional computing discipline", "maintain" leaves no doubt that it is at least *in part* such a profession (as well as, in part, a "discipline" within "computing"). When the ACM Executive Council approved SWECC's charter (and did not appoint Shaw to the Steering Committee's successor), it in effect declared that Shaw had lost the argument over the *professional* "maturity" of software engineering.[10]

The Annual Report listed six "Key Events" for 1998, the first two of which occurred in June. The ACM and IEEE-CS agreed to work with the Texas Board of Professional Engineers to "identify a technical basis for recognizing competency in software engineering" and "[enact] rules that recognized software engineering as a distinct engineering discipline." In August, the NCEES (National Council of Examiners for Engineering and Surveying) passed a resolution in support of "licensing software engineering throughout the nation." These first three events, so closely bunched, may help to explain why Shaw had lost the argument (for the moment at least). Whatever the ACM or IEEE-CS thought, events seemed to be quickly turning software engineering into a part of engineering. The only question still open seemed to be how it would be done. The next three "key events" offered a tentative answer. In September, the Industry Advisory Board for the Guide to the Software Body of Knowledge Project "held its first meeting in Mont Tremblant, Quebec". The body of knowledge task force (soon to be known as "SWEBOK") was hard at work on the "Guide" that licensing bodies would need if they were to offer examinations in software engineering. In October, the engineering accreditation agency (ABET) and its computer-science counterpart (CSAB) "announced the integration of their accreditation services". And, on November 11, the ACM and IEEE-CS jointly hosted a meeting to "initiate the update of Computer Curricula 1991". Led by software engineering, computer science seemed to be moving closer to engineering.[11]

The Steering Committee's last report ended with two "Project Results" in keeping with these events. The first was that the ACM and IEEE-CS had approved the software engineering code of ethics. (Like the Guide to the Body of Knowledge, the Code provided a basis for distinguishing between those who should be licensed as software engineers and those who should not.) The second "project result" was that the Curriculum task force had completed its "model accreditation criteria for software engineering" in November and that the ACM and IEEE-CS "education boards" had endorsed them. Though the Curriculum task force had started much more slowly than SEEPP, it had finished at almost the same time. When the Steering Committee officially dissolved at the end of 1998, it could take considerable satisfaction in what it had accomplished. Of its four original assignments, it had completed two, defining "ethical standards" and "educational curricula" (or, at least, standards for such curricula). A third (defining the body of knowledge) was well on its way to completion (though it had proved a much bigger job than expected). Only the fourth, "adopting standard definitions", had been abandoned—and only because the Steering Committee had decided that the IEEE already provided an adequate definition of "software engineering" and related terms.[12] The Joint Steering Committee might (like General Wolfe on the Plains of Abraham) "die content".

12.2 SWECC takes over

On January 28, 1999 (in Austin, Texas), SWECC held its first meeting, a nine-hour marathon beginning at 8:30 AM, with only twenty minutes for lunch. All six members were present (though Northrop was only present by a proxy, Laurie Werth, a resident of Austin).[13] The minutes, nine single-spaced pages, are full of interesting summaries of past work and plans for the future. The minutes have six attachments: "History of SWECC"; "Overview of Software Engineering Standards"; "Report on SWEBOK Project"; "Overview presentation given to Texas Board of Prof. Engineers"; "Overview of SWECC"; and a "White paper by Dennis Frailey on cooperation". (All but the last attachment consist of several pages of overheads.) All these documents (as well as the Steering Committee's Annual Report) were placed on the web. (Tripp seems to have liked the sun to shine on whatever he was doing.)

We may draw three conclusions from an examination of these early SWECC documents. The first is that Frailey and Tripp, but especially Tripp, seem to be the source of most of the activity. For example, two of the attachments are Frailey's ("History" and "White paper"); the other four seem to be Tripp's. The minutes identify five "Responsibility Areas". Frailey has one ("Education"); Reed and Northrop share one ("Publicity"); and the other three are Tripp's ("SWEBOK", "Liaison with Standards", and "Liaison with National Bodies").

The second conclusion to draw from the minutes of January 28 is that the Body of Knowledge is not only Tripp's focus but also SWECC's (just as it had been for the Joint Steering Committee) and, as they believed, integral to the Curriculum project (now "Education"). Much of the morning half of the Austin meeting seems to have been informational (with the first session devoted to "Overview and History of SWECC Project" and the second to "Relationships between SWECC and others). The last session of the morning is, like the first three afternoon sessions (each an academic hour), for "Plans". But then there is a short session (twenty-five minutes) on "Software Engineering Standards" followed by a long session (fifty-five minutes) for "Report on SWEBOK Project"—with "Responsibility Areas" and "Future Plans" completing the day. There was, it seems, no special session for Education, even though all three highlighted resolutions of the day explicitly mentioned the "Education Project". SWECC adopted three resolutions at the Austin meeting. Two of the three mention Education along with SWEBOK. The third, while it does not explicitly mention SWEBOK, at least implicitly relies on its results: the undergraduate curriculum is to be the "priority" for the Education Project and that curriculum is to include "the core content of the software engineering field."

This description of what SWECC did at its first meeting suggests the third conclusion I draw from the minutes. The Code of Ethics was not on the agenda at all. It was, however, not entirely forgotten. It is mentioned once in the minutes—the last item on a list of four "proposed projects" (the other three of which are "Guide to SWEBOK", "SWE Model Curriculum", and "SWE Performance Norms").[14] The Code of Ethics was also mentioned in several of the overheads. Whatever the Code of Ethics was, it was no longer "a problem".

The next face-to-face meeting was to be in Montreal a half year later (July 13, 1999), apparently because the technical experts working on the Body of Knowledge were in Montreal (at the University of Quebec). SWECC also scheduled a "teleconference during the week of

April 12, 1999." It was, it seems, to be a short meeting, allowing SWECC to evaluate progress on various "action items".

12.3 Back in Tennessee

Meanwhile, Gotterbarn was trying to put the Code into the hands of as many software engineers as possible. These efforts took three forms: publication, consultation, and organization. Typical of the first was the publication in the January 1999 issue of *Software Engineering Notes* (an ACM SIGSOFT journal) of Version 5.2 along with a brief introduction under the title "A Positive Step Toward a Profession: The Software Engineering Code of Ethics and Professional Practice". Though the article was typical of Gotterbarn's attempts to publicize the Code, response to it was not. A letter to SEN's Editor objected both that the Code was "too vague" and that it added "nothing to the existing codes of ethics for engineers". The time had come, it asserted, "to move beyond 'motherhood statements' and try to give meaning to the title, 'Software Engineer', by establishing standards that restrict the use of the title to a proper (and probably small) subset of those who now use it."[15] The Editor notified Gotterbarn of the letter as soon as he received it, offering to print a response beside it in the next issue (May). For Gotterbarn, what began as a simple article had turned into a crisis. The letter's author, David Parnas, was (and remains) one of the "greats" of software engineering, especially among software engineers concerned with professional ethics.

The importance of Parnas to software engineering was evident in the very issue of SEN in which his letter to the editor appeared. A few pages after the letter is an "ACM Fellow Profile" of him, including both a short biography and a three-page interview. A licensed Professional Engineer in the Province of Ontario, with a BS, MS, and Ph.D. from Carnegie Mellon (all in Electrical Engineering), Parnas held the NSERC/Bell Industrial Research Chair in Software Engineering at McMasters University. He was the author of more than two hundred papers and reports, recipient of the ACM Best Paper Award in 1979, of two Most Influential Paper awards from the International Conference on Software Engineering (ICSE), and (in 1998) of the ACM SIGSOFT Outstanding Research Award. Those interested in software ethics revered him as one of the early opponents of Reagan's plan to defend against nuclear attack from space ("Star Wars"). Parnas resigned from a federal advisory panel in protest because "as a professional…I must devote some of my energy to deciding whether the task that I have been given is of benefit to society". He argued that the Star Wars project could not do what it was supposed to do, could not be adequately tested even if it could, and would in any case not be reliable enough to be worth the expense. He made his arguments in public, stressing that his resignation was not simply a "personal" decision but one any responsible professional in similar circumstances should take.[16]

Gotterbarn's response to Parnas' letter was four times as long as the letter itself. It was authored not by Gotterbarn alone, but by Miller and Rogerson as well, with "SEEPP Executive Committee" (rather than their academic affiliations) under their names (even though neither SEEPP nor its executive committee any longer existed).[17] Apparently, Gotterbarn thought he needed to speak with as much authority as possible. He also had to treat Parnas with respect, while explaining how wrong he was. The response justifies the level of abstraction the code adopted (what Parnas calls "vagueness") by arguing that the Code is just abstract enough not to use standards that are (and are likely to continue to be) "in flux". A code of ethics should not, for example, require “mutation testing for all mission critical software”, though it should require “adequate testing” (however a software engineer should understand that term at the time). A code of ethics should be abstract enough not to be outdated in a few years.

A code of ethics should not, however, be so abstract as to be altogether “uncontroversial” —and the Software Engineering Code is demonstrably not that abstract. Here the IEEE Standards process provided evidence of how much controversy there was about the Code:

> In 149 pages of comments developed during the IEEE-CS formal review, the following conflicting comments were made by practicing “professional” software developers: add the clause “avoid criticism of software engineers’, “remove 1.08 …volunteer professional skills to good causes”, “…make 1.08 mandatory”, “remove references to maintenance”, “include all forms of maintenance”, “…not all jobs require the highest professional standards” [and so on].[18]

Gotterbarn attributes this variety of opinion to “software engineering [being], at best, an emerging profession”. That, I think, is a mistake. Diversity of opinion on such questions is, as far as I can tell, characteristic of all professions, even of law and medicine, and has always been, perhaps because any substantial number of human beings tend to have a wide range of opinion, unless forced to reach agreement by circumstance. What distinguishes the “fully developed professions” from those that are merely “emerging” is simply a core on which most (but not all) agree. In this respect at least, Gotterbarn’s data seems to show that software engineering was, by 1999, no longer an emerging profession. The Standards process had documented something close to ninety percent agreement on the Code as a whole (and *no* opposition to *most* provisions).

Gotterbarn might have added that the error of considering a code of ethics “uncontroversial”, “trivial”, or otherwise no more than common sense when it is not, is an “honorable error”, one most likely to be made by those who hold themselves to high standards. Assuming that others think as they do, they cannot see the point of the code’s requiring no more than they require of themselves. Those who work to a lower standard tend to make the opposite mistake, that is, to suppose that the code is “unrealistic” or “impractical”, sets too high a standard”, and so on. They are often surprised to discover that few of their colleagues agree with them about what is practical. The open discussion of what is “uncontroversial” and “impractical” itself helps to shape the discipline’s understanding of both. We have no direct insight into much of practice beyond our own and perhaps that of a few people with whom we work closely.

Below Gotterbarn’s response to Parnas was a related news item. The ACM had established an Advisory Panel on Professional Licensing in Software Engineering to assess “the current role of ACM as part of a joint effort with the IEEE Computer Society to oversee and actively participate in the development of professionalism and licensing guidelines through [SWECC]”, to recommend “an appropriate role for ACM to take with respect to these issues”, and to provide a “brief final report to the ACM Council by its next meeting [May 16. 1999]”. Among the fourteen members of this advisory panel was David Parnas.[19] While the committee was “free to determine its course”, the Council expected it to “consult with the representatives of the areas of ACM that have an interest in the topic [including]: SIGSOFT (David Notkin, Chair), the Ed Board (Peter Denning, Chair), and the ACM representatives to SWECC (Dennis Frailey [and the others]).” Licensing was again causing trouble, more than Gotterbarn could then guess.

The Gotterbarn-Parnas exchange is just one example of how Gotterbarn sought to disseminate the Code through publication.[20] Typical of attempts to disseminate it through consulting is an exchange of emails with Dell Computer's legal department ("Dell Legal"). Late in January, a paralegal at Dell Legal wrote Gotterbarn. That email is lost, but Gotterbarn's January 22 response makes it clear that he understood Dell to be asking permission to copy the Code. He informed the paralegal, "The copyright is registered to me", expressed pleasure at Dell's interest, and asked what exactly Dell had in mind.[21] A few days later, the paralegal explained that her "team is working to enhance our existing guidelines for procurement professionals to be appropriate for a wide group within Dell". Her team would like to adopt some of the Code's "concepts" but it "would not be a direct adoption". She then offered to send Gotterbarn a draft if "it would make you more comfortable". The draft of "1/26/99" ("Standards of Business Ethics for PSG Professionals/Engineering Code of Conduct") arrived a few days later (February 16). Nine pages long, it was an impressive document, but showed no obvious borrowing from the Code.

Gotterbarn's statement that the Code's copyright is "registered to me" deserves comment. Actually, the copyright was in the name of Miller and Rogerson as well as Gotterbarn. The copyright in question seems to have covered all versions of the code, starting with "version 1.0".[22] Copyrighting the Code was part of a strategy, dating from September 18, 1998, to avoid two dangers. One was that the short version of the Code might be printed without the long. The other danger was that the ACM or IEEE-CS might charge for the Code (as they did for other documents), thus (unintentionally) discouraging dissemination and use, or bury the Code by failing to grant permission to copy it. By copyrighting the Code, Gotterbarn avoided both dangers. He could provide free downloads of the Code at the SEERI website on condition that the Code not be divided and that no charge be made for copies. Neither the ACM nor IEEE-CS was happy with this arrangement, however. They viewed the Code as a "work for hire" and therefore theirs, not Gotterbarn's, even though no one who had had any part in writing the code had been hired (except the ACM grammarian and IEEE-CS lawyer). By the middle of 1999, Gotterbarn, ACM, and IEEE-CS had resolved the dispute. The two societies became joint holders of the copyright, but only after agreeing that the Code should carry below the copyright symbol the statement: "This Code may be published without permission as long as it is not changed in any way and it carries the copyright notice." It carries that notice to this day.[23]

Copyrighting the Code was a small part of efforts to create the organizational infrastructure to disseminate the Code more widely. The Software Engineering Ethics Research Institute (SEERI) was another (10. 2); so too was ACM's Committee on Professional Ethics (COPE). But once SWECC replaced the Joint Steering Committee, it seemed obvious that dissemination had to include SWECC, especially if SWECC got the funding and staff support it was asking for. Gotterbarn therefore developed a plan for a "project" under SWECC corresponding to the task force he had chaired under the Steering Committee. Unfortunately, by this time, the Software Engineering Education Project had the acronym "SWEEP". So, Gotterbarn could not name the project he was planning "the Software Engineering Ethics Project". Its obvious acronym, SEEP (not to mention "SWEEP"), would be too easy to confuse with the Education Project's SWEEP. Gotterbarn dismissed "the Software Engineering Ethics and Professional Practices Project" (SEEPPP or SWEEPPP) for the same reason. Eventually, he chose "SEPEP" (Software Engineering Professionalism and Ethics Project), with the lucky suggestion of energy ("PEP"). Consistency with the nomenclature of other SWECC names would entail a “W” after the “S”, but Gotterbarn thought that consistency not worth the loss of the imperative sounding “SE[E]”.

Gotterbarn submitted a plan to SWECC’s meeting on April 15. SEPEP’s plan had four parts: Purpose, Structure and Governance, Immediate Projects, and Initial Committee Membership.[24] SEPEP’s purposes included publicizing the Code, fostering broader acceptance of the Code across international borders, updating the Code “as appropriate”, and working with SWEEP to incorporate into software engineering education a knowledge of the Code and its use in decision making. Among SEPEP’s immediate projects were developing a flyer about the Code to be included with all ACM and IEEE-CS membership renewals, presenting a panel on the Code at appropriate ACM and IEEE-CS software engineering meetings, publishing the Code in *Communications of the ACM* and *Computer*, working with the Education group to develop educational materials on the Code, and maintaining a joint ACM/IEEE-CS website on the Code. SEPEP was to have a three-member executive committee. Its chair was to be “a member of both the ACM and the Computer Society”, with the other two members representing (respectively) the ACM and the IEEE-CS. The chair would report to SWECC annually. Though SWECC (in January) had required all projects to have a definite duration, SEPEP’s plan included no termination date (though it did provide for a fiscal year beginning July 1). Nonetheless, the plan was approved “as presented”.[25]

The meeting at which this happened (a teleconference) lasted only seventy-five minutes. SEPEP had fifteen of those minutes. Ahead of SEPEP was SWEEP’s plan. The chair (Tripp) presented it “[on] behalf of Jerry Engel and Richard LeBlanc”. SWEEP had forty-five minutes. After SEPEP was “new business”, a fifteen-minute discussion of “opportunities for promoting the Guide to the Software Body of Knowledge Project”. All the “action items” for the meeting concerned the Body of Knowledge. All six members were “present”.

12.4 A happy but short life

When SWECC next met (Montreal, July 13, 1999), only four members were present (Ardis, Carver, Frailey, and Tripp). At another all-day meeting, they approved “Guidelines for Projects and Proposals for New Projects”, the minutes of the January and April meetings, and a proposal to conduct annual surveys of international software engineering programs. They also approved the content of a “Code of Ethics poster”, though with the suggestion that “the presentation could be improved by consulting with a graphics artist”. Perhaps most important, however, SWEBOK reported that the “Stoneman” version of the Guide to the Software Body of Knowledge (version 0.5, that is, something well short of Version 1.0) was nearly complete, with the remaining two Knowledge Areas expected to be complete by July 23. A “few hundred” reviewers had volunteered, covering all the Knowledge Areas, but more reviewers were being sought “to represent other points of view”. Beside the Texas Board of Professional Engineers, “other groups (especially in Canada) have been taking actions about licensing software engineers”. There was agreement that SWECC “needs to enter into formal agreement for the exchange of materials” with all of these groups. SWECC would meet again in person in late January 2000 and, if there was enough business, by teleconference before then, in October.

SWECC did not meet again, even by teleconference, until March 4, 2000. But it nonetheless ended 1999 with a major achievement, a whole issue of *IEEE Software* (November/December) devoted to it, almost sixty pages of text. Tripp (along with Steve

McConnell) was Guest Editor (with an Introduction to the issue, pp. 13-18). David Parnas contributed the lead article (pp. 19-30), "Software Engineering Programs Are Not Computer Science Programs", a spirited and intelligent justification for training software engineers as if software engineering were just another kind of engineering. Parnas even set out—course by course—the curriculum he would prescribe for an undergraduate software engineering student. Immediately following Parnas was Gerald Engel's report on SWEEP, "Program Criteria for Software Engineering Accreditation Programs" (pp. 31-34). While SWEEP's undergraduate criteria were designed to augment the computer science criteria ("CASC/CSAB"), the graduate criteria tracked ABET's engineering criteria. Following Engel's paper was "The Guide to the Software Engineering Body of Knowledge" (pp. 35-44). Three of its five authors (Piere Bourque, Robert Dupuis, and Alain Abran) were from the University of Quebec at Montreal. The other two were James Moore (MITRE Corporation) and Tripp. The Stoneman version of the Guide was available on the web for comment. Other versions would follow, each taking into account comments on the one before. The Body of Knowledge would always be a work in progress. Moore had an article of his own about another SWECC project, developing "An Integrated Collection of Software Engineering Standards" (pp. 51-57). Gotterbarn followed Moore, explaining "How the New Software Engineering Code of Ethics Affects You" (pp. 58-64). This article includes two inserts: one, the short version of the Code (without the long); the other, a list of "Early Adopters". Among the early adopters is McConnell's Construx Software (and the United Kingdom's Royal Mail).

The remaining three pieces concern licensing. The first of these ("What Do You Mean I Can't Call Myself a Software Engineer?") was in the middle of the issue (pp. 45-50). Its author, John Speed, was at the time a member of the Texas Board of Professional Engineers (and himself a licensed—civil—engineer). His article is largely descriptive, explaining what the new Texas regulations of the term "software engineer" require. The other two pieces (and a short "Responds", at the end of the issue, are officially polemical (being grouped under the title "Head to Head"). Dennis Frailey defended licensing (pp. 66 and 68). Tom DeMarco, a principal in a "New York and London based consulting group", argued against. For DeMarco, the market provides "the most efficient and elegant certification mechanism ever envisioned" (pp. 67 and 69). The only alternative to the market, a "Soviet-style central licensing bureaucracy", would do only one thing, protect "a class of senior practitioners whose skills are so irrelevant and so far out of date that they can't really do any highly paid job except licensing." Frailey's response to this free-market attack on licensing (p. 69) is to point out that DeMarco does not know what he is talking about. Licensing professionals is almost universal, even in countries without a "Soviet-style bureaucracy". And because licensing regulations allow the licensing body to impose conditions for keeping a license, "licensing is one of the few mechanisms that encourages and, in some jurisdictions, even requires continuing education." Licensing raises standards beyond what the market would otherwise impose.

Licensing was still an issue likely to rouse passions, perhaps as likely to rouse passions as it was in 1993 when Buckley included licensing in his IEEE-CS Board of Governors motion to begin establishing software engineering as a profession (2.3). Evidence of how licensing could still arouse passions appeared in the February 2000 issue of the *Communications of the ACM.* The ACM President, Barbara Simons, wrote a "Viewpoint" piece, "Not Now, Not like This", stressing that a licensing test "will [not] assure that the person who passes the test will be qualified to write programs that will never endanger the public." Part of the reason a test cannot provide that assurance is that we "do not have building codes for programs." Indeed, we do not even have "a vocabulary of program design rich enough to discuss structural integrity."[26] Simons had two co-authors, Fran Allen and Paula Hawthorn, both members of the ACM advisory panel established in March 1999 to report on licensing software engineers.[27]

That panel had reported to the ACM Executive Council on May 15, 1999 (on schedule).[28] The Viewpoint piece was, in effect, disseminating the four conclusions of that report—which the Council had endorsed. The Council had declared that the "ACM opposed" licensing software engineers "at this time" because it believed that licensing would be "premature" and would not address "problems of software quality and reliability". The ACM was nonetheless "committed to solving the software quality problems", but (the Council declared) the way to do that was "by promoting R&D, by developing a core body of knowledge for software engineers, and by identifying standards of practice." While the ACM would continue to advise "boards and other agencies interested in licensing of software engineers", its "interaction will not imply ACM's endorsement of licensing or involve associated materials such as tests." Last, the Council undertook to form an "ACM SWECC Oversight Committee" to oversee "the development of the software engineering body of knowledge and ACM's participation in SWECC." This report *seemed* to favor most of what SWECC was doing.

The first and last appendices to the May 15 report provided further insight into how contentious licensing remained. Appendix A, by John Speed, is the official report of the Texas Board of Professional Engineers (approved February 4, 1999). A crucial element of the Texas approach was that a licensed "software engineer" had to be an engineer (strictly so called): "Not everyone with computer related experience is eligible to become a licensed Texas P.E. by applying under the software engineering discipline." The Board would decide "on a case-by-case basis who has been practicing engineering". This requirement was not as hard for computer scientists to satisfy as it might seem. The Board had a definition of "the practice of software engineering" that made explicit reference to "computer sciences". More likely to be discriminatory was the requirement that an applicant have letters of reference from "at least 9 people, 5 of whom must be licensed engineers."

Did computer scientists actually have anything to fear from engineers? Appendix C of the Report at least suggested that they did. According to that document, the Senate of the Memorial University of New Foundland, Memorial's academic governing board, had approved an honors program in the Department of Computer Science with a specialization in "software engineering". Upon learning of this degree, the Association of Professional Engineers and Geoscientists of New Foundland had joined the Canadian Council of Professional Engineers (CCPE) to *sue* Memorial over the use of the term "engineering". The case was set for hearing in the federal court in St. John's in September 1999.[29] The Association of Universities and Colleges of Canada (representing most of the degree-granting institutions in Canada) pledged to cover eighty percent of any legal costs Memorial incurred. What began as a simple academic decision had, because of the word "engineering", become an expensive federal case.[30]

On March 4, 2000, SWECC (again) met in Austin.[31] Northrop and Carver were absent, but the other four members were present, along with Gotterbarn (for the "Ethics Project"), Laurie Werth ("observer from Education Project"), Dupuis and Bourgue ("for SWEBOK"), and Gerald Engel (present for "the first part of the meeting to report on the progress of the [Education Project]" through his representative, Laurie Werth). The "Summary of Resolutions" gives the impression that the Ethics Project was the central concern of this meeting. Of four resolutions, three concerned ethics. SWECC (now written "SWEcc") approved a logo for the code of ethics, approved copyrighting the logo as a "committee asset", and approved "the proposed revised budget for the code of ethics project and flyer distribution".[32] Another resolution listed in the Summary thanked "the SWEBOK team for the quality of its work and its contribution to software engineering as a profession." Though these resolutions together suggest a meeting both dull and short, the minutes themselves (four pages, with one page-long appendix of "Action Items" and another of "Acronyms") suggest something else.

The first important item of the day concerned proposals in Indiana, New Mexico, and Missouri to license software engineers. The proposal from Missouri was "particularly important" (so important that each SWECC member received a copy). It would treat software engineering as a specialty within computer engineering. The proposal would be submitted to the NCEES (National Council of Examiners for Engineering and Surveying) in Fall 2000 as an alternative to the Texas proposal. There was a "general perception that the Missouri proposal would be less desirable than the Texas proposal" not only because it could "divide the [computing] community but also [because it placed] software engineering in a less influential and fundamentally inappropriate position as a subsidiary of computer engineering in the minds of the professional engineering community." After considering possible responses, SWECC took no action. SWECC responded in the same way to the Texas Board's request to "review the definition of software engineering".[33]

After this half-page discussion, the minutes of SWECC's March meeting offered Gotterbarn's report on the Code of Ethics Project, a half-page report on an informal survey of software engineering programs, an informal report on SWEEP (Education), and a half-page report on SWEBOK (Body of Knowledge). Since July 1999, the Guide to SWEBOK had completed its "stoneman" phase and, despite "some relatively minor schedule slips due to delayed responses by certain authors", was close to completing the "ironman". SWECC planned no further face-to-face meetings in 2000. It would, however, have at least one conference call with SWEEP and SWEBOK "to review progress" and another conference call (even later) to "discuss approval of the SWEBOK final document".

That March meeting was SWECC's last. Three months later (June 30, 2000), the ACM's Executive Council voted to withdraw from SWECC, in effect killing the organization.[34] SWECC had survived little more than eighteen months, less than a third as long as the temporary Joint Steering Committee for which it was supposed to be the permanent replacement. The Executive Council that voted to withdraw from SWECC was the same one that had approved entry. It withdrew just before leaving office. To make clear that it did not take its decision lightly, the Executive Council issued a six-page explanation ("A Summary of the ACM Position on Software Engineering as a Licensed Engineering Profession"), with three appendices: A) two pages of "Q&A" summarizing the Report; B) a ten-page "Assessment of the Software Engineering Body of Knowledge Efforts"; and C) a twenty-page report "On Licensing of Software Engineers Working on Safety-Critical Software".[35] Two of the three authors of the Body of Knowledge report should be familiar: David Notkin *and* Mary Shaw.[36]

SWECC (according to the Summary) had "increasingly been perceived as furthering efforts to license software engineers." Two particular efforts stood out: "1) communications with the State of Texas relative to Texas' desire to develop a licensing exam for software engineers under the general framework for professional engineers (PE's); and 2) the development of a first-draft of the body of knowledge (SWEBOK) that all software engineers would be expected to know." It was such efforts that led the Executive Council to appoint a "blue ribbon panel of prominent software engineers" to investigate "the issue of software engineering and the need for licensing". The Council had decided in May 1999 that "it could not support licensing software engineers." The state of knowledge and practice in software engineering is "too immature to warrant licensing". Licensing would, under the circumstances, be "ineffective in providing assurances about software quality and reliability." In May 2000, the Council concluded as well that "the framework of a licensed professional engineer, originally developed for civil engineers, does not match the professional industrial practice of software engineering." Because SWECC had become "so closely identified with licensing of software engineers under a professional engineer model, the ACM Council decided to withdraw from SWECC." Though the ACM had withdrawn from SWECC, it continued to believe that "the problem of reliable and dependable software, especially in critical applications, is the most important problem facing the IT profession." ACM would continue to "work closely with [IEEE-CS] on projects to further the evolution of software engineering as a professional computing discipline and improve the quality of software and the capabilities of software engineers."

There are, I think, two (compatible) ways to read what ACM did. One is that ACM had no objection to software engineering becoming a profession; the objection was to software engineering becoming an integral part of engineering (rather than a "professional computing discipline" or an "IT profession").[37] Integration into engineering would have important consequences for computer science as an academic enterprise. What consequences? Early in 1999, the Naval Academy had announced a graduate program in software engineering within Computer Science (and Information Systems), the first in the US.[38] Later that year, Carnegie Mellon University had announced a similar program. The CMU announcement listed a faculty of four, all from the School of Computer Science (Institute for Software Research International).One of the four was Mary Shaw.[39] The graduates of such programs might well have trouble getting licensed as software engineers in Texas. If the Texas approach spread, programs like that at the Naval Academy and CMU might die—and computer science programs be the poorer in consequence.[40]

The other way to read the ACM decision is as a misunderstanding. That was how Frailey read it. Writing for himself in the *Forum for Software Engineering Education* (FASE), he "strongly disapprove[d]" of the ACM decision; it would have "detrimental long term ramifications to software engineering and to the ACM"; and it put "in doubt" the work of "over 400 volunteers from about 50 countries" for "weak and often incorrect" reasons.[41] Relationships "with key external organizations" that had taken eight years to build had been severed with the loss of good will and respect for the ACM. Frailey was equally disappointed with how the ACM had reached its decision. The process had been "exclusionary". Neither the Executive Council nor its task forces had "consider[ed] the views and insights of ACM's appointed SWEcc representatives…the people actually doing the work—even after we offered to provide information and to assist in the discussions". One result was that the draft rationale contained "incorrect assumptions and factual errors" that, though easily corrected, "may well have influenced the Council's decision." For example, the draft report said that SWECC "focused…on matters of licensing", it did not—"ask anyone who has attended one of our meetings". SWECC had not, as suggested, supported licensing in ways inconsistent with ACM policy. In fact, it had "dissuaded the Texas Board from developing an examination and convinced them that ACM was an appropriate authority to turn to for advice in this matter." While Frailey hoped that "the work of the four SWEcc projects will continue under some banner" supported by the "software engineering community", the work would now have to go on without "the coordination and communication functions that SWEcc provided, the united front presented by the ACM and the Computer Society working together, and the influence on these activities that ACM could have had." The ACM had, in effect, ceded the project of establishing software engineering as a profession to IEEE-CS—and IEEE-CS had always looked on licensing with more favor than the ACM had. For Frailey, ACM's withdrawal from SWECC was a strategic blunder.

12.5 Software Engineering Ethics Coordination after SWECC (2000-2004)

This book's importance depends (in part at least) on the importance of the process studied. If writing the Software Engineering Code in particular, or establishing software engineering as a profession in general, is merely a failed attempt to establish a profession, the thinking about professions preserved here will seem less important for understanding professions. This book would seem likely to tell us no more about living professions than an autopsy of a fetus spontaneously aborted can tell us about children born alive. We are, of course, not yet in position to know that software engineering has succeeded in becoming a profession. We need a decade or two—at least. Events between SWECC's dissolution and completion of this book do, however, provide some evidence that the attempt is succeeding. I can think of no better way to conclude this chapter than by (briefly) recounting that evidence.

Software engineering has continued its transformation from a field of computer science into a field of engineering (strictly so called). Two events seem especially relevant here. On March 14, 2001 (after what he described as "2+ years" of silence), Gotterbarn emailed a score or so of those involved in writing the SE Code of Ethics that (among other things) the "IEEE has started a Certification for Software Engineering Professional." In addition to "experience, education and passing a test they require adherence to the Code".[42] When I looked at the IEEE-CS website in 2004, I found only a certificate program for "Software Development Professional", but it did include a link to the SE Code of Ethics.[43] The Code remains an integral part of IEEE certification of software development professionals (if not exactly of software engineers), apparently the only IEEE-CS (or IEEE) certificate program. Whether or not certification is essential to profession, it is clear that the certification of software engineers is now treated as part of engineering certification—and linked to the SE Code.

Sometime in August 2001, the ACM's Task Force on Licensing of Software Engineers Working on Safety-Critical Software issued a "Final Report". The Report opposed licensing (for reasons already familiar) but endorsed "codes of professional conduct" (without mention of the Software Engineering Code by name).[44] The Task Force did, however, recommend that the ACM review its codes of ethics from time to time. While the ACM does not seem to have taken this recommendation to heart, the IEEE-CS has.

On November 7, 2002, the IEEE-CS Professional Practices Committee (Leonard Tripp, Chair) concluded that "the Code is stable and currently there is not sufficient evidence that it is in need of revision."[45] The Committee reported only two criticisms of the Code. One, apparently widespread, was that the Code did not "outline specific sanctions for violations". The Committee thought this no more than a misunderstanding: "Those sanctions are generally contained in the bylaws." The Code need only, and does, "refer to potential sanctions…[in] 8.09" ("Recognize that personal violations of this Code are inconsistent with being a professional software engineer"). The other criticism received was not at all common: "One Florida based computer manufacturing company's lawyer said the company could not adopt the Code because Clause 6.06 admits the possibility of a circumstance where someone may ['in exceptional circumstances'] have to violate a law to protect the public." The company adopted all the other clauses of the code. The Committee thought the Code was right to say that the law should not trump the public interest.

The report's appendix listed some evidence of the Code's success: five textbooks in "Software Engineering" that either reprinted the Code or posted it on the book's website; nine textbooks in "Computer Ethics" that made similar use; seven companies that had adopted the code (and several more that posted it on their website); and adoption of the (short version) of the Code by the Association of Information Technology Professionals.[46] Most impressive, perhaps, were the number of translations posted on the web, three "official" (Chinese, Italian, and Spanish) and three more "unofficial" (Croatian, Hebrew, and Japanese). An official French translation was underway (for French Canadians).[47] (A translation becomes official when Gotterbarn, after consultation with software engineers who speak both languages, concludes that the translation is reasonably faithful to the original.)

On April 10, 2004, Keith Miller emailed Simon Rogerson (with copies to Gotterbarn and me) reporting that he had recently attended a workshop (in California) on software testing: "I was pleased to hear several speakers mention the joint ACM/IEEE code at the workshop when the issue of professionalism came up." Though the "lack of enforcement rules was mentioned", it was also "mentioned that the code was an important first step for the profession." The "technical world" seemed to know about the code and that knowledge seemed to "make a difference".[48]

Meanwhile, even the Guide to the Software Engineering Body of Knowledge (SWEBOK)—apart from licensing, the most controversial part of the professionalization project—continued to develop. On September 15, 2001, Dupius, Bourque, and Abram, the three Canadians who had taken over "editing" SWEBOK, issued an interim report. "Trial Version 0.95" (just short of Version 1) had been published in February 2001 for review. In April 2001, the project's Industrial Advisory Board "recognize[d] that due process was followed in the development of the Guide (Trial Version) and endorse[d] the position that the Guide (Trial Version) is ready for field trials for a period of two years." In May 2001, the IEEE-CS Board of Governors accepted SWEBOK (Trial Version) "as fulfilling its development requirements" and also declared it ready for two years of field trials.[49]

A little over a year later (July 25, 2002), Bourque emailed a general call for position papers and participation in a workshop on "improvements to [SWEBOK] and to the Software Engineering Education Body of Knowledge (SEEK) to be held October 6, 2002 at his home institution, the Ècole de technologie surpierieure, in Montreal. The results of the workshop were, in turn, to be used as "input to a second (to be confirmed) workshop" at the Conference on Software Engineering and Training (to be held in Madrid in March 2003). While SWEBOK now belonged entirely to IEEE-CS, SEEK was still an "ACM/IEEE Computer Society initiative to define undergraduate software engineering curriculum recommendations." On May 29, 2003, Bourque issued a call for volunteers to review Version 1.0—almost the final step for publication as a Technical Report by the International Organization for Standardization (ISO TR 19759). On February 6, 2004, the IEEE-CS Industrial Advisory Board unanimously agreed that "the Software Engineering Body of Knowledge project initiated in 1998 has been successfully completed; and endorse[d] the 2004 Version of the Guide to the SWEBOK and commend[ed] it to the IEEE Computer Society Board of Governors for their approval." The IEEE-CS Board then approved the "2004 Edition of the Guide" (and authorized the Chair of the Professional Practices Committee to proceed with printing).[50] SWEBOK will hereafter be reviewed (and revised) like any other IEEE standard.

Software engineering now has an (internationally defined) body of knowledge, accreditation guidelines (both graduate and undergraduate), and a code of ethics. It has procedures in place to maintain (and improve) the relevant documents, to teach them to the next generation of software engineers, and even to integrate them into current practice. Assuming these achievements are neither mirages nor ephemera, what more does software engineering have to do to count as a profession (strictly so called)?

NOTES

[1] Annual Report – 1998, Joint IEEE Computer Society and ACM Steering Committee for the Establishment of Software Engineering as a Profession. www.acm.org/serving/se/Rpt9811 (3/23/2004). This is, as far as I can tell, the only annual report the Joint Committee ever issued (or, at least, ever placed on the web). The web version seems to have first appeared in November 1998 ("9811") and was last updated on "2 March 1999", but (according to the draft SWECC schedule of September 29, 1998) should have been prepared in February 1999. (Gotterbarn\ Version 5\ACMSUPPORT\SCHEDV20). Since Frailey's name precedes Tripp's, the order of names is unusual. Usually, the chair's name would be listed first. While Tripp might be listed second simply in submission to the alphabet, the more likely explanation seems to be that the order recognized that Frailey did more of the work.

[2] Gotterbarn\Version 5\ACMSUPPORT\ACMREPS (file date: 9/29/98). Ardis is now a professor in the Department of Computer Science and Software Engineering, Rose-Hulman Institute of Technology, suggesting (as many other biographies here have as well) how porous is the boundary dividing software engineering's "technical community" from its "educational community".

[3] www.acm.org/serving/se/mi990128 (4/11/2004). By January 1999, Tripp had replaced Carver as IEEE-CS President. The people are, it seems, much more stable than what they "represent".

[4] Gotterbarn\Version 5\ACMSUPPORT\CHARV22 (file date: 9/29/98). The document itself indicates that it is as "approved on 19 October 1998" by the ACM, suggesting that it might be more recent. But I doubt it. The document also claims to be as approved by the IEEE-CS "XX November 1998". The final document (available at www.acm.org/serving/se/Charter, 4/11/2004) differs from it in a number of (minor) ways. My guess is that, on September 29, Frailey and Tripp knew the exact date of the ACM Executive Council meeting (only three weeks away) but not that of the IEEE-CS Board (almost two months off). They had merely "penciled in" the future. What we have here is very much a draft.

[5] Here is one place where the final Charter differs from the one in Gotterbarn's files. The language of Gotterbarn's draft was "an intended duration".

[6] SWECC's charter is also interesting for what it has to say about resources. The two societies agree to provide funding for SWECC, dividing expenses evenly, except that IEEE-CS is to "provide a secretariat service for SWECC". www.acm.org/serving/se/Charter (4/11/2004). Another draft document of September 28, 1998 is a budget for "FY 1999 and 2000", calling for a total of $12,600 for the first year and $25,000 for the second. Gotterbarn\Version 5\ ACMSUPPORT\BUDGET2. There is no budget on line.

[7] "Software Engineering Professionalism and Ethics Project (SEPEP), April 1999". www.acm.org/serving/se/Sepec (3/23/2004).

[8] Gotterbarn/History of SE code/ History of proj.

[9] www.acm.org/service/se/History (2/23/2004).

[10] Compare the “Rationale” for SWECC offered to ACM and IEEE-CS. Gotterbarn\Version 5\ACMSUPPORT\Report.

[11] ABET has long included “technology” programs as well as engineering programs in its accreditation work and, from the ACM’s perspective, integrating CSAB with ABET may have seemed no more than a more efficient way to do what it had been doing all along.

[12] www.acm.org/service/se/History (2/23/2004).

[13] www.acm.org/serving/se/min990128 (4/11/2004).

[14] “Performance norms” seem to be another way to talk about “standards of professional practice”.

[15] David Lorge Parnas, “On Code Reuse”, *Software Engineering Notes* 24 (May 1999): 4

[16] David Lorge Parnas, “SDI: A Violation of Professional Responsibility”, *Abacus* 4 (—— 1987): 46-52. Reprinted in (among other places): Deborah G. Johnson, editor, *Ethical Issues in Engineering* (Prentice Hall: Englewood Cliffs, New Jersey, 1991). Quote in Johnson, p. 15.

[17] Don Gotterbarn, Keith Miller, and Simon Rogerson, “On Code Reuse: a Response”, *Software Engineering Notes* 24 (May 1999): 4-6.

[18] Gotterbarn, Miller, and Rogerson, p. 5.

[19] The others were: Fran Allen (co-chair), Barry Boehm, Fred Brooks, Jim Browne, Dave Farber, Sue Graham, Jim Gray, Paula Hawthorn (co-chair), Ken Kennedy, Nancy Leveson, Dave Negal, Peter Neumann, and Bill Wulf. *Software Engineering Notes* 24 (May 1999): 6. This is a committee of notables, but of academic notables [explain].

[20] For other examples of Gotterbarn’s printed dissemination of the Code in 1999, see: Don Gotterbarn, “Two Approaches to Computer Ethics”, *SIGCSE Bulletin* 31 (June 1999): 11-12; Don Gotterbarn, Keith Miller, and Simon Rogerson, “Software Engineering Code of Ethics Approved”, *Communications of the ACM* 42 (October 1999): 102-107; and Don Gotterbarn, Keith Miller, and Simon Rogerson, “Computer Society and ACM Approves Software Engineering Code of Ethics”, *Computer* 32 (October 1999):84-88.

[21] Gotterbarn\Version 4 IEEE Vote\SEPEC after 5.1\Dell\Shari1.

[22] Gotterbarn\Version 4 IEEE Vote\SEPEC after 5.1\Copyrigh. [sic]

[23] Gotterbarn once remarked in conversation that he now regrets failing to make notification a condition of use. With notification, he would have a better idea of how widely used the code is.

[24] www.acm.org/serving/se/Sepec (3/23/2004). Another version of this plan, indistinguishable except for its title date of “December, 1999”, is on the IEEE-CS website: www.computer.org/tab/SWEC9912 (4/11/2004).

[25] www.acm.org/serving/se/min990414 (4/11/2004). SWEEP’s plan, adopted the same day, has a fifteen month schedule (but no “termination date”). www.acm.org/serving/se/sweep (4/11/2004).

[26] Barbara Simons, Fran Allen, and Paula Hawthorn, “Not Now, Not Like This”, *Communications of the ACM* (February 2000): 29-30.

[27] The inclusion of Hawthorn among the co-authors provided some balance. While Allen, along with ten other members of the panel, had opposed licensing, Hawthorn, along with two others (Browne and Parnas), had favored it.

[28] “ACM Panel on Professional Licensing in Software Engineering Report to Council (May 15, 1999)”. www.acm.org/serving/se_policy/report (4/22/2004).

[29] For more details, Karen Hawthorne, “Engineering program accreditation to proceed at Memorial University”, *Engineering Dimensions* (May/June 1999): 10.

[30] In addition, CCPE halted the process of accrediting Memorial’s engineering programs—until the Supreme Court of New Foundland told CCPE it had gone too far. It could not hold up the accreditation of engineering programs as a way to pressure Memorial to reverse its decision concerning a computer science program.

[31] This meeting was held in conjunction with the 13th annual Conference on Software Engineering Education and Training, March 6-8, 2000, also in Austin.

[32] www.computer.org/tab/present/min20000304 (4/18/2004).

[33] The Texas definition was quite detailed, but (by allowing for “computer sciences”) did not obviously exclude most ACM members likely to be interested in licensing: "The practice of software engineering will mean a service or creative work such as analysis, design, or implementation of software systems, the adequate performance of which requires appropriate education, training or experience. Such education, training or experience shall include an acceptable combination of: computer sciences such as computer organization, algorithm analysis and design, data structures, concepts of programming languages, operating systems, and computer architecture; software design and architecture; discrete mathematics; embedded and real-time systems; or other engineering education. Such creative work will demonstrate the application of mathematical, engineering, physical or computer sciences to activities such as real-time and embedded systems; information or financial systems, user interfaces, and networks." www.acm.org/serving/se_policy/report (4/22/2004).

[34] ACM members were not notified of the decision until a “Summary” was published on the ACM website (www.acm.org/serving/se_policy/selep_main) on July 7, 2000 (and made final on July 17). Don Bagert, “ACM Withdraws from SWEcc”, FASE (*Forum for Advancing Software engineering Education*) July 2000. www.cs.ttu.edu/fase/v10n07 (4/18/2004), p.11.

[35] www.acm.org/serving/se_policy/selep_main (April 22, 2004). The authors of the report on licensing were: John Knight, Nancy Leveson, Michael DeWalt, Lynn Elliot, Cem Kaner, and Helen Nissenbaum. The final version of this report (dated August, 2001) seems to have been published (or, at least, “last modified”) on October 12, 2001, bearing the copyright “@2001 John Knight and Nancy Leveson, all rights reserved”.

[36] The third author, Michael Gorlick, had worked at Aerospace Corporation for more than twenty years on such projects as “wearable computing, grid computing, peer-to-peer infrastructure, large-scale data mining for structural genomics, internet appliances, and wireless devices. Sunset.usc.edu/gsaw/bios/gorlick (April 23, 2004).

[37] The ACM did not make clear in what sense software engineering either is or might be a “profession”. Insofar as software engineering had a code of ethics of its own (and seemed to be trying to live by it), it was by this point a profession separate from engineering—whether or not it was a “computing discipline” or an “IT profession”.

[38] FASE, July 1999. www.cs.ttu.edu/fase/v9n07 (4/18/2004).

[39] FASE, February 2000. www.cs.ttu.edu/fase/v10n02 (4/18/2004). This “Institute” is not to be confused with the Software Engineering Institute. According to its website (http://www.isri.cs.cmu.edu), “[i]n addition to participating in the School of Computer Science's top-rated undergraduate program in Computer Science, ISRI offers: A PhD Program in *Software Engineering*, A PhD Program in *Computation*, *Organizations and Society*, Professional Masters programs in *Software Engineering* and *Electronic Commerce* Distance education in *Software Engineering* with and without academic credit.” Italics in original.

[40] Indeed, the problem for computer science might be even worse in some states, those that require anyone teaching “engineering design” to be a licensed engineer.

[41] FASE, July 15, 2000. www.cs.ttu.edu/fase/v10n07 (4/23/2004), p. 14.

[42] Email (Gotterbarn to Davis), March 14, 2001.

[43] http://www.computer.org/certification/cert_for_you.htm (August 6, 2004).

[44] John Knight et al, “On Licensing of Software Engineering as a Profession: Final Report of an ACM Task Force, August, 2001”, http://www.acm.org/serving/se.policy.html (February 8, 2003). While the report also endorsed curriculum improvement as a way to improve software, it expressed considerable doubt that the Software Engineering Body of Knowledge (SWEBOK) would become a useful document any time soon. Apparently, SWEBOK was still linked in many

ACM minds with licensing. See, for example, J. Barrie Thompson, "Any Real Progress, or Is it Just Politics and Turf Wars?" FASE (September 15, 2001), p. 18. www.cs.ttu.edu/fase/v11n09 (5/21/2004), p. 14 : "The real difficulties are obviously associated with the Texas model for licensing software engineers and it is really unfortunate that the SWEBOK project is viewed by many (perhaps wrongly) as being inextricably linked to the Texas actions."

[45] "Professional Practice Committee Meeting 11/7/2002". For explanation of this report, see Gotterbarn's email to me (April 28, 2004): "IEEE revises technical documents—the IEEE professional practices committee asked me to report to them in 2002 on the status of the Code, I have attached draft of that report". The report is an attachment to this April 28 email. The Committee can only be found on the IEEE-CS website at http://www.computer.org/csinfo/ standingcoms.pdf (September 20, 2004). The Committee, though apparently predating SWECC, seems to be acting as its successor (within IEEE-CS, of course). Its agenda for November 7 certainly resembled SWECC's. There is, for example "liaison" with the "SWE Curriculum project", the "SWEBOK Guide (Ironman Version)", and "SWE licensure activities" (among others). The Committee had ten members (beside Tripp). One of them was Dennis Frailey. FW PPC #42 7 NOV 2002.

[46] The report lists the following website as its source for AITP adoption: http://www.aitp.org/downloads/AITPAdoptsthe SoftwareCode.doc. When I checked that site on August 11, 2004, it contained no reference to the SE Code.

[47] On April 28, 2004 (email to Davis), Gotterbarn reported a German and an Arabic translations "in progress". All "official" translations are posted on the website of Gotterbarn's Software Engineering Ethics Research Institute: http://seeri.etsu.edu/Codes/default.shtm (August 15, 2004). All seven of the translations mentioned in the 2002 report are there.

[48] On July 18, 2004 (email to Davis), Gotterbarn announced in a postscript: "Did I tell you that the Australian Computer Society and the Australian Institute of Engineers have adopted the Code[?] That is especially interesting because it gives it a 'semi'-legal status. Following it is part of your professional obligation. I have been having discussions with the New Zealand Society and they are likely to adopt in a few months."

[49] *Forum for Advancing Software Engineering Education* 11 (September 15, 2001), p. 3. http://www.cs.ttu.edu/fase/v11n09.txt. By then SWEBOK had been examined by almost "500 reviewers in 40 countries" (during one of three reviews). Bourque email, May 29, 2003.

[50] http://www.swebok.org/home.html (August 11, 2004).