Chapter 11: The Long Process of Approval, 1998

"Whenever you set out to do something, something else must be done first."
—Murphy's Law #12

11.1 An unpleasant surprise

After all the activity of 1997, Gotterbarn enjoyed the relative calm of the first few weeks of 1998. Only a few Code-related emails, like stragglers after a great battle, drifted in from time to time. The first arrived on January 7. Its author, a sixth-year student in computer science at the Technical University of Eindhoven, had a page and a half of "opinions and criticisms" concerning Version 3 (9. Appendix). While he liked most of the Code, he thought it "geared towards software engineers working in a (large) company." As an "independent software engineer", he "personally" disliked "this point of view". Since most codes of engineering ethics have been criticized for not paying enough attention to engineers working in large organizations (large organizations being where most engineers work), the Dutch student had, however unintentionally, complimented the Gotterbarn.

The Dutch student also thought the Code felt "very US-oriented". Several "political[ly] correct terms" obscured "the meaning for us foreigners". Had Version 4 taken care of that problem (10. Appendix)? For example, the old "diversity" clause (renumbered 1.07) now read: "Consider issues of physical disabilities, allocation of resources, economic disadvantage and other factors that can diminish access to software." No "politically correct terms" there. Perhaps Version 4 had taken care of another problem as well.[1] The Dutch student was concerned that some clauses "may be in violation of local laws". The only example he gave was "5.05" ("Develop a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which an employee has contributed"). Version 4 had dropped that clause; perhaps it had done the same with others the Dutch student might have mentioned.[2] This email from the Netherlands seemed to endorse what the executive committee had done in Baltimore.

Much the same was true of a much longer (six page) email, equally thoughtful, that arrived the next day from Switzerland. Its author, George Sigut, held a position at the Federal Institute of Technology in Zurich much like that Manny Norman held at Eastern Michigan. In charge of Computer Services for the Institute, he was a software engineer practicing within an academic (research) environment.[3] Sigut proposed: "1 Changes in the form [;] 2 Changes in the Principles [; and] 3 Other changes". One of the changes in form, though different from what Gotterbarn, Miller, and Rogerson had made in December, was surprisingly similar. Sigut proposed that, instead of Version 3's two-part division into Preamble and Principles (with Clauses), the Code should have a three-part division: Preamble; Principles; and Clauses. He wanted the reader to be able "to consider the Principles on their own [without a major effort]". Version 4's "short version" made that possible. Like some other critics, Sigut also wanted to distinguish clauses according to the three categories described in Version 3's Preamble (Aspire, Expect, and Demand). Gotterbarn, Miller, and Rogerson had considered doing that too, but had instead abandoned explicit mention of the categories to avoid confusion. Perhaps Sigut would

have agreed that deletion of the distinction was an equally good way to resolve the problem he had identified. Why not?

Sigut had given the Preamble considerable attention. His suggestions for changes constitute about a quarter of the entire six-page email. Their general drift is to sharpen the distinction between Principles and Clauses (a distinction fundamental to the design of the Code yet a source of misunderstanding ever since Version 1). For example, his "Point 5" is to amend the fifth paragraph by deleting "Principles and" from the sentence, "The list of Principles and Clauses is not exhaustive." The reason for this change is (he suggested) obvious: "Principles are exhaustive by definition". He was, in part, suggesting a reversion to the language of Version 1 ("no set of subsidiary clauses exhausts the general rule [Principle]"). But he also seemed to be making a larger point about the Principles, something Version 1 did not. The Principles defined the special responsibilities of software engineers (what moral obligations they have beyond what everyone else has). The Principles might, if badly designed, leave something out that should have been included. But what was left out would be something that *should* be a special responsibility, not something that *is*. That is the sense in which the Principles are exhaustive "by definition". They were supposed to be exhaustive (whether or not they in fact were). The clauses were not like that. They were just instances of the Principles. There could always be more instances. They were *not* supposed to be exhaustive.[4]

Sigut also picked out for revision many of the clauses Gotterbarn, Miller, and Rogerson had already deleted, for example, old 6.07 ("Only accept remuneration appropriate to professional qualifications or experience"). Here and there, though, Sigut identified a problem Gotterbarn, Miller, and Rogerson had not. For example, Sigut says of old 7.06 (also new 7.06) that the wording might be improved. It then read: "Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and security measures in general." Would it not, Sigut asked, read better if rewritten to say: "assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files, security measures in general, and other confidential information"? Apparently, Sigut thought "and security measures in general" looked tacked on to a well-ordered list. He suggested inserting it in the list (without the "and"); there would then be only one (longer) list and only one "and" instead of two.[5]

Gotterbarn now had a set response for late comments on Version 3. It began, "Dear Enlightened Person" (until the individual's name was inserted). After thanking the enlightened ones for "your thoughtful response", it informed them, "We have already factored in numerous responses to version 3 of the code and have produced version 4.0." Version 4 (a copy of which was printed below the letter) "is now up before the IEEE-CS/ACM Steering Committee." Nonetheless, "[your] comments will be fed into the ongoing maintenance and development of the code and its associated accessories (teaching packs and so on)." The results of "this effort" are to be available "early this year" at the ACM and IEEE websites (and at the website for Rogerson's center). The names of the three members of SEEPP's executive committee appear below the final "Sincerely".

Gotterbarn replied on the same day he received Sigut's email (January 8). Though he might have sent Sigut the standard letter, apparently he thought Sigut deserved better. Addressing him as "Dear George", Gotterbarn inserted two sentences in the middle paragraph of the standard reply: "The three level distinction in the preamble drew endless debate about which

clause went at which level. We removed the distinction from the Preamble and (in general) put the more aspirational elements at the top."[6]

A few days later (January 11), another correspondent proposed a new principle or clause (apparently designed to cut across existing Principles):

> Perform factually-based work analyses and estimations to define validatable and verifiable systems requirements, then only budget adequate resources for hard requirements or seek approvals for reduced scopes of work based on hard budgets prior to specifying, accepting and assigning the work.

The author of this disastrous sentence had a legitimate problem he was trying to solve. Many projects "fail because they have a fatally flawed set of requirements or a contradictory mission and budget." They are wasteful because of "some preliminary character flaws of the planners and upper managers". The software engineer should therefore be responsible for evaluating the initial specifications, revising or rejecting them as appropriate, not just accepting them in the way a "servant" would. Who would disagree? But why had the author of this email not noticed that Version 3 (like Version 4) had in fact dealt with the problem? 0ld 1.04 (like new 3.02) required software engineers to: "Ensure proper and achievable goals and objectives for any project on which they work or propose."[7] Had the author of this email read the Code? Perhaps not. There was nothing in the email to show that he had.

This commentator plainly deserved nothing more than the standard letter (with perhaps a reference to the clause that already did what he suggested). But another plainly deserved a response even more personalized that the one Gotterbarn sent Sigut. That commentator had sent an email only a half page long, but its author, Stuart Zweben, an ACM-appointed member of the original Steering Committee (and retiring ACM president). Gotterbarn agreed that Zweben had "identified two of the sore spots in version 3.0." But, in "early December we did a significant revision of the [Code]." Zweben had not liked the wording of old 3.07 ("Refuse to participate in decisions of a governmental or professional body…") or old 2.07 ("Not put self-interest, the interest of an employer…"). Gotterbarn quoted the wording of the corresponding provisions of Version 4 (4.06 and 1.02), suggesting that the new wording had taken care of the problems Zweben identified. Clause 4.06 allowed participation but only after disclosure of any conflict of interest; 1.02 required only that the software engineer "moderate" the interests of self, employer, and so on with the public good. Gotterbarn concluded by noting that he had "tried to address your other question in my 19 January letter to the Steering Committee". He also informed Zweben that "Version 4.0 is available at www.acm.org/serving." Gotterbarn seems to have sent this response to Zweben on January 19.[8]

Zweben's email implicitly told Gotterbarn something that worried him. When he sent Version 4 to the Steering Committee (that is, to Cabrera and Frailey) on December 16, Gotterbarn was anticipating an adoption process rather like that by which the ACM had adopted its code of ethics. The Steering Committee would promptly approve sending Version 4 to the two societies. The Board of Governors of the IEEE-CS would meet in February, consider Version 4, and approve it (perhaps with a few amendments); the ACM's Executive Committee would do the same about the same time. Gotterbarn might have to go before the IEEE-CS Board or the ACM Executive Committee to explain this or that; he might have to meet privately with individual members of one or the other body; but it would all be over by March. By the first day of Spring,

1998, Version 4, or something very like it, would be the official Software Engineering Code of Ethics and Professional Practice; Gotterbarn could turn his full attention to its dissemination—and to teaching, to his Institute, and to his other projects. But that schedule required the Steering Committee to vote on the Code soon after receiving it. For the last month, Gotterbarn had been assuming that, though the Steering Committee had not responded to him, it was at least working on his recommendation, the holiday season explaining any delay in taking the official vote. What Zweben's comments on Version *3* told him was that the Steering Committee was not working on the recommendation of Version 4. Indeed, it told him something worse: Cabrera and Frailey had not, during the intervening month, even told the rest of the Steering Committee that Version 4 existed. Why not? Why had Frailey rushed SEEPP to complete work before the end of the year if he was not even going to pass the Code on to the Steering Committee's members in December?

To find out, Gotterbarn emailed the entire Steering Committee, not just its chair and vice chair, a letter similar in substance (but not organization) to the letter by which Gotterbarn had originally transmitted Version 4.[9] The first paragraph announced that SEEPP had sent the Steering Committee a recommendation in December and had "not heard any response" since.[10] The second paragraph worried about timing: "I believe the Board of Governors of the IEEE-CS and the Executive Committee of the ACM have meetings scheduled in February and I would be willing to represent the SEEPP task force at these meetings." Having the Code approved at those meetings would be "salubrious [because] there are already several events scheduled for February and March which include the Code."[11] Gotterbarn "fired [this] second letter to the whole committee 20 January."[12]

By January 27, Gotterbarn had learned enough to write Miller with details "[just] to keep you up on the fun".[13] Both Zweben and (Dennis) Frailey had responded to Gotterbarn's "second letter". Zweben was "sorry [but] the ACM EC met last week and the next meeting in the US is in May." In other words, the Steering Committee's failure to act quickly had put the ACM schedule for approval back at least three months. That news was distressing enough, but Frailey's was worse. Frailey was, Gotterbarn told Miller, "our only real contact with the committee". Cabrera, having become involved with a major project at Microsoft soon after becoming chair, had become increasingly distant from the Committee's work (much as Gotterbarn's co-chair, Melford, had two years before). Though only vice-chair, Frailey had been trying to hold the Steering Committee together. He was not doing well. He "did not know the status of the steering committee". The quotation marks are Gotterbarn's around Frailey's exact words. Gotterbarn reported replying: "co-chair [sic] did not know status of steering committee????!!!!!" Frailey explained that "no one on the committee is responding to his emails." (The Committee had long since ceased having face-to-face meetings; and, Gotterbarn concluded, even telephone conferences had ceased many months ago.) Frailey was almost ready to "take unilateral action". Here Gotterbarn added his own smile symbol and continued, "We have some experience with this form of in-action from the Steering Committee." So, Gotterbarn was wondering whether he should suggest that Frailey email the Committee saying that, there being no objection, he would forward the recommendation to the two societies by Valentines Day, "the first completed deliverable from the Joint Steering Committee." Gotterbarn would, in other words, suggest "our mode of operation—if he hears no dissent from his committee, Dennis should forward the Code and recommendation to both societies." Gotterbarn ended this chuckling note with a serious

question: “What do you think of this as a recommendation to Dennis? Any suggestion on how he should frame his proposal to the Steering Committee to minimize potential dissent?”

Gotterbarn must have reached Miller on one of his dark days. He responded that Gotterbarn could suggest the silence-means-consent strategy “but I doubt that it would be accepted. I think they WANT to do a pocket veto.” Miller did not explain who “they” were (Cabrera and Frailey, some other members of the Steering Committee, or the Steering Committee as a whole) or why “they” would want to do a pocket veto. Whoever “they” were, Miller thought the better strategy was to go around them: “request, in strong terms, that you or the whole exec. Committee be invited to the May meeting and that we get some guaranteed time on the agenda.” It may be the time “to meet this thing head on instead of out-maneuvering them.” Gotterbarn took Miller’s advice, though he waited till April to do it. Meanwhile he tried to find out more about the status of the Steering Committee, the likely reception the Code would receive at the May meeting of the ACM Executive Council if the Steering Committee remained dormant, and the likely reception at the May meeting of IEEE-CS Board of Governors under similar conditions.

11.2 Spring maneuvers

On March 1, Amr El-Kadi (the professor in Egypt) wrote Gotterbarn asking for an update “on the response you get from the Steering Committee on V 4.0 of the Code”. El-Kadi wanted to “know how that ends.” He also wanted to know about “the other two task forces: have they completed their work” and “in what form would the steering committee merge the work of the three task forces?” Gotterbarn did not respond until March 16 (apparently using the two weeks intervening to learn what El-Kadi wanted to know). The response, a page long, provides a good summary of the state of the Steering Committee (and its work) at that time. Though addressed to the listserv, it preserved El-Kadi’s email at the bottom and began by acknowledging “the following questions from Amr”. What followed immediately was the wisdom of an experienced committee chair: “Life is not simple.” Gotterbarn then reported that on December 16 he sent out “our recommendation” that Version 4 be forwarded to both “the IEEE-CS and ACM,” that “both societies had meetings of their executive committees in late January and early February”, that he had received no response from the Steering Committee to the December recommendation, and that Version 4 had *not* been forwarded when he checked in January. He had sent a “follow-up letter to the Joint Steering Committee” in January and later “telephoned the ACM co-chair [Dennis Frailey]”. Frailey had told him that the Steering Committee “had not yet formally responded to my (our) request(s).” The Committee had not even responded to Frailey’s requests. More phone calls revealed that the Steering Committee’s “IEEE-CS chair [Cabrera] has resigned”![14] By the time Gotterbarn learned of Cabrera’s resignation, it was the middle of February and both the IEEE-CS Board of Governors and the ACM Executive Committee had met and “we were not on their agendas”. Gotterbarn “spoke with the President of the IEEE-CS in March”; she had “asked someone to be the new IEEE-CS co-chair”.[15] Gotterbarn was still trying to find out when the next Board of Governors meeting would be, but had been put on the agenda of the next ACM Executive Committee meeting in the US—in May, in Washington, DC.

That was the simpler part of Gotterbarn’s administrative life. The email now explained some of the complicating factors. At least “part of the problem we are having” seemed to be “related to the status of the education and skills task force products”. The “skills task force”

(Body of Knowledge) had "spent a lot of money" but had not finished its work. "Some people" wanted to hire a professional group to finish it. Meanwhile, the "education task force" (Curriculum) had been "waiting on the skills task force results, but I think that has currently changed—along with the leadership of the education task force." Gotterbarn then directed anyone interested to a website with reports from all three task forces. (The reports there, except for SEEPP's Version 4, were then all almost a year old.)[16] The change in leadership might not end the Curriculum task force's troubles, Gotterbarn added. The "U.S. based accrediting agency—ABET" had asked the *IEEE* (but not IEEE-CS) to "define a software engineering curriculum". So, it seemed, "there are two distinct efforts going on related to curriculum."[17] The Steering Committee seemed to need more than a new chair.

Before promising to "keep trying to work through the steering committee", Gotterbarn reported that he was "doing my best to publicize the Code." He had had some successes. The Code had been "adopted by some industries" and will soon "appear in some text books". It is "on a web site in Bosnia" and will "appear in some publications". He was hoping it would become "a defacto Standard, [so] that it will be easier for it to get support from the societies."

While Gotterbarn was willing still to work through the Steering Committee, he was no longer willing to work through it alone. He had to be prepared for the Committee again to fail to do anything. So, on April 8, he addressed a formal letter—"Dear Charles House [ACM President] [18] and Dennis Frailey".[19] The letter "respectfully requested[ed] that [Version 4] be considered at the next Executive Committee meeting in May for adoption by the ACM." A similar request was going to the IEEE-CS in time for the June meeting of its Board of Governors.

That was only the beginning of a letter that went on for more than a page. Its second paragraph explained that SEEPP had sent the Steering Committee Version 4 in December and followed up with another letter in January. (SEEPP's letters of January 19 and December 16 were appended in case there was any doubt about what they said.) "We understand," Gotterbarn continued, "the Steering Committee's response has been delayed, owing to reorganization." That delay had "several unfortunate consequences". One was that the "computing community has criticized this joint effort as going nowhere" when in fact there was one, if only one, "finished product", the Code. Second, the Steering Committee's efforts were in danger of trailing events. At a recent conference at which licensing software engineers in Texas was discussed, the Code "was described as a significant contribution to professionalization and licensing of software engineers." Would it not be unfortunate if the Texas Board adopted the Code as their standard before it was approved by the sponsoring societies"? Third, the Code was already receiving "international recognition". That recognition included adoption "by industry such as Siemen's Software Development, India"; the Code's publication in several computer ethics text books ("including Kevin Bowyer's revision of his IEEE-CS book on Computer Ethics"); discussion of the Code at several professional meetings; and its forthcoming publication "in the Journal of Business Ethics, Journal of Science and Engineering Ethics, the Proceedings of the ACM Policy Conference in May (in Washington, DC), and the SIGCSE Bulletin."[20] Would it not be unfortunate, Gotterbarn seemed to ask if, with all this support for the Code, the Steering Committee's inaction were allowed to stop the ACM and IEEE-CS from endorsing it?

The letter then offered three reasons, each having its own paragraph, for the ACM (and IEEE-CS) to endorse the Code. First, the Code had "undergone extensive review by those who will be affected by its adoption" as well as "by ethics officers of large industries, philosophers, and ethicists". Version 4 "represents a general consensus". Second, failure to endorse would

indicate “a lack of support for ethical practices”, forcing “each software practitioner to re-invent the morality of software engineering”. Third, the ACM should “take a stand to provide all software engineers with an ethical direction and with a tool they can use to educate others—including their management—about the ethical responsibilities of the software engineer.” The Code did what the ACM wanted it to do, “clearly state the moral commitments of practicing professionals and of the profession.”

A few hours after sending off this letter, Gotterbarn had a response—from Frailey.[21] He wished that “you had contacted me before sending this [because the] steering committee is preparing a recommendation to the two bodies you mentioned.” That was good news—even if tardy—and more followed. There was “consensus, at least there appears to be, that the code of ethics part [the short version] is suitable for recommending.” Frailey then got to the bad news: “there is some concern that the professional practices part [the main code] is not ready for recommendation, mainly because some steering committee members feel the field is too immature to be recommending such practices.” That news raised many questions. It was Mary Shaw (like Zweben, one of the original ACM-appointed members of the Steering Committee) who typically argued from the premise that software engineering was “too immature” a discipline to be a profession. She had been arguing from that premise for almost a decade. Now, even with support at 95% or above for most provisions of Version 3 (and therefore, probably, for most provisions of Version 4 as well), she had, it seemed, no second thoughts. She was still talking as if there were no consensus on standards of practice. She had, however, (apparently) not remained entirely unmoved. She was now willing to accept the “code of ethics” (although the vote on Version 3 had not included it, that is, the “short version”); what she was still not willing to accept was the “code of practice” (the main body of the code on which a clause-by-clause vote had been taken). What part, if any, did evidence play in her decision? The opinion of practicing software engineers? The “immaturity of the field”? Was she alone in her opposition to the full Version 4? Was Frailey again with her? Did she have other allies? (“*Some* steering committee members” sounded like more than one, but perhaps Frailey wished to give that impression simply to protect the confidentiality of deliberations.)

The Steering Committee was (Frailey continued) “trying to work this out” and planned to “make a formal recommendation to the two councils in May.” Frailey had only been “waiting to contact [Gotterbarn] further until more members of the steering committee had participated in the discussion of the committee’s final recommendations.” What did that mean—“more members”? The Steering Committee was a not large body. It had ten members—not counting the ex officios (who were, apparently, not to be part of these deliberations). Or, rather, it had no more than nine members until Cabrera was replaced. If Frailey was (as he said) still trying to get more members to participate, the consensus he described must be among only a part of the Committee, perhaps a small part, a fragile consensus and one in which even one articulate member (such as Shaw) might carry great weight. There was, then, still no guarantee that the Steering Committee would do anything in time for the May meeting. Frailey was in effect telling Gotterbarn that the Steering Committee was barely operating—and, therefore, that going around it had been wise. Gotterbarn’s letter to the ACM might protect his time on the May agenda; it certainly had made it harder for the Steering Committee to do nothing or to send just the short version of the Code forward. Everyone at the ACM Executive Council would know that SEEPP had recommended the entire Version 4, not just the short version, and had done that in December (almost five months before the May meeting).[22] Gotterbarn had, it seemed, out-maneuvered the

Steering Committee—or at least a minority of it. There might be awkward questions at the Executive Council meeting, but at least the complete Version 4 would be in front of them in May. He could make his apologies then.

Having offered Gotterbarn some hope, Frailey returned to a subject about which Gotterbarn had worried since September 1997, the dissolution of the Steering Committee. Frailey posed (what he called) "one simple question": "Should the code of ethics activity continue and, if so, what bodies of ACM and the computer society should sponsor it?" The Steering Committee was going to recommend that each of its sub-activities "be continued, if appropriate, to be jointly sponsored by the appropriate bodies in the two societies, with a new 'coordination' body to be formed as a replacement for the steering committee". Of course, Frailey's question was not simple. The coordinating body could be a permanent version of the Steering Committee under Barbacci, active and helpful, or a presence even more shadowy than it had become under Cabrera, or any number of other things—better or worse than the Steering Committee had been. Everything depended on the details: membership, budget, powers, reporting line, and chair. Though Frailey offered no details (and seemed to have none), Gotterbarn could see that the Steering Committee's dissolution might not be the disaster he had anticipated. He would have to think hard before he answered Frailey's "simple question".

A few days later, Gotterbarn learned that Leonard Tripp had replaced Cabrera as Chair of the Joint Steering Committee. Though Tripp had been a member of the Steering Committee since early 1996, Gotterbarn had never met him. (The Steering Committee had held only one face-to-face meeting during Cabrera's term (November 1996), holding that one without ex officios—except for Melford.) Gotterbarn knew Tripp only from one phone conversation back in 1994. Melford had called Tripp, one of the "IEEE-Standards gurus", during the April 24 meeting in DC, to ask about how to write a "PAR" (3.4). A distant voice coming through a speaker phone set in the middle of the conference table, Tripp had seemed a true "organization man".[23]

Had Gotterbarn learned more of Tripp during the intervening years, he might have found much to confirm that first impression. Though a member of the ACM as well as of the IEEE-CS, Tripp had largely confined his professional activities to the IEEE. He had served on the Operations Committee of the IEEE-CS Technical Council on Software Engineering, chaired the IEEE-CS Standards Activities Board, Standards Coordinating Committee, and Software Engineering Standards Committee. He also chaired the US Technical Advisory Group on software engineering standards. Had Gotterbarn known all this, he would have had no trouble understanding why Melford considered Tripp "Mr. Standards". He might, however, have wondered why Tripp took such an interest in Standards. The explanation could not be Tripp's education. Both his degrees were in mathematics (Brigham Young University, BS and MS). As a mathematician, Tripp *should* have gravitated toward ACM activities (and away from Standards) rather than toward IEEE-CS (and Standards). What seems to explain his preference for the IEEE-CS—and for standards—is his employer. Tripp had worked for the Boeing Company for thirty years. Boeing is an engineering organization *par excellence*, building some of the world's largest, most sophisticated aircraft, both civilian and military. Technical standards, including standards for software, are central at almost every step of what Boeing does. Tripp had (like many other software engineers) moved from thinking of software as mathematical formulas to thinking of it as the most complex of machinery, ethereal but vital. His specialty had become standards for safety-critical airborne software. For him, the Standards-setting process was a matter of life or death.[24]

Gotterbarn worried about having "Mr. Standards" in charge of the Steering Committee. For him, the term "IEEE Standards" meant meetings concerned with PARs, CFPs, the Operations Guide, Scopes, and other procedural matters that bored him, produced nothing but procedural documents, and seemed to draw nothing but yawns from ordinary volunteers. Gotterbarn recalled Chikofsky's 1994 warning that the IEEE Standards process could go on for "a long time" (3.4). Since that warning, Gotterbarn had learned that "a long time" meant "years"; he had himself slaved three years in the salt mines of that complex process.[25] When Melford resigned in December 1996, Gotterbarn felt as if chains had been struck from his legs. He had abandoned the IEEE Standards process, replacing it with something like the simple process by which the ACM had adopted its code. He had accomplished more, infinitely more substantively, in one year in that way than he had in the three preceding years following the IEEE standards process. Was all that had been achieved in 1997 to disappear into the endless tunnels of that dark process?

Since Tripp was now in charge, there was nothing to do but to try to work with him. So, a few days after learning of the Steering Committee's new chair, Gotterbarn wrote Tripp "a status report" clearly designed to show him that the IEEE process had, in effect if not in form, already been satisfied. "We" worked on the code for "almost two years" (counting, it seems, Mechler's efforts in 1996 as well as his own in 1997, but not 1994's procedural activities). There had been "a broad based development effort—truly international in participation—with contributions from a wide range of practitioners." Having framed the process in this way, Gotterbarn took a page to summarize what was already in the December and January letters of transmittal to the Steering Committee and in the March letter to the ACM (the reasons to move quickly). That letter is dated April 20.[26]

Early Sunday morning, April 26, Gotterbarn sent Mechler the standard email informing him that (in response to his report of a new email address) he had (once again) been added to the PRFCMP-L mailing list. A few minutes later Gotterbarn added a personal note (apparently, to answer Mechler's "what's new?"). "[Trying] to manage a list at another site" is, Gotterbarn observed, "no fun", but Mechler had not missed any messages. There was news, though. There had been "some leadership changes on the steering committee". Gotterbarn was "working to get the Code version 4 in front of both the IEEE-CS and ACM executive committee meetings." Then, changing the subject, Gotterbarn asked whether Mechler knew "of anyone using the Code." Gotterbarn knew "of one Seimen's shop, an insurance company, a division of Hitachi, one army training program and that is about all I remember at this moment."[27] Mechler seems not to have responded.

Two days later (April 28), Gotterbarn sent an official letter ("Dear President House") requesting the ACM Executive Council to consider adopting Version 4 at its May meeting. The letter exists in two electronic forms. One is ordinary text; but the other is on the letterhead of Gotterbarn's Software Engineering Ethics Research Institute.[28] Using software just becoming available, Gotterbarn had made his electronic mail as impressive as possible. He sent the letter and a copy of Version 4 (as he said in a covering memo) as "support material for the ACM's meeting in May". The letter itself indicated he did this on Frailey's instructions.[29] Though much the same as his earlier letters of transmittal, this one included one new item with the title "Software Alliance". The Texas Board of Professional Engineering Licensing Committee was, he reported, "considering the Code for their proposed licensing standard for Software Engineers". Following that startling news, he listed the Code's recent successes: It was being

used in a program for industrialists in Australia, at Army facilities at Fort Monmouth (New Jersey) "in training materials for computer engineers, computer scientists, and computer specialists", and so on.

On May 5, Gotterbarn heard from IEEE-CS. After several days of "telephone tag", he and its new president, Doris Carver (co-chair of the Curriculum task force, 1994-96), had finally been able to talk. She later sent an email confirming the substance of the conversation (with a copy to Tripp).[30] The "procedure you will need to follow is for the report to be submitted via the steering committee". The IEEE-CS half of Gotterbarn's attempt to bypass the Steering Committee had failed. Carver pointed out (what she—correctly—thought Gotterbarn knew already) that Tripp was the new Steering Committee chair. "[The] best path is", she advised, "for the two of you to communicate." Gotterbarn decided to wait until he knew what the ACM Executive Council would do on May 10. If they approved the Code, he would be in a better position to work out something with Tripp.

By the time Gotterbarn attended the ACM's Executive Council Meeting on May 10, he knew that he had underestimated the Steering Committee (or, at least, Tripp). No longer a "silent partner", it too had put an item on the Council's agenda.[31] Though titled "Report of Task Force on Software Engineering—Software Engineering Code of Ethics & Professional Practice", the item was in fact a status report on the work of the Joint Steering Committee. John Werth (Carver's ACM counterpart on the Curriculum task force, 1994-96) spoke for Frailey (who was absent). After listing the four purposes for which the Steering Committee had been established in 1993, Werth reported that the Engineering Licensing Committee of the Texas Board of Professional Engineers had approached the Steering Committee in 1997 for assistance with "the issue of software engineering licensing".[32] The Licensing Committee had "formally requested that ACM and IEEE-CS jointly work with it to define a body of knowledge on which national engineering licensing exams [for software engineering] will be based." The Texas Board of Professional Engineers would next meet in Fort Worth, June 17-18, and might then enact rules for licensing software engineers in Texas. The question for the Council was what to tell the Texas Board at their June meeting. The Steering Committee knew at least as much about events in Texas as Gotterbarn did.

The minutes report one Council member or another observing that "coming out as a discipline will be a challenge" for software engineering, that there "will be a need for strong industry representation", that it was "not known whether industry was driving this issue", that the Texas Board had already set criteria for licensing disciplines "less well defined than SE", and that software engineering had in recent years changed from a "supporting activity" to a "princip[al] player." When the discussion ended, the Council unanimously took two actions. First, it declared that the "ACM would be pleased to assist the Board in defining the body of knowledge and standards of practice on which such licensing exams would be based [should the Texas Board vote to adopt rules for licensing software engineers]." The ACM would also be "pleased to work with the IEEE CS in this endeavor." Second, the Executive Council requested that "Dennis Frailey or Leonard Tripp, in consultation with other members of the Steering Committee and, if possible, including participation from SIGSOFT and the ACM Education Board, propose a management structure whose responsibility would include: 1) coordinating follow-on activities of the Steering Committee in the body of knowledge, ethics, professional practices and education areas; 2) assisting State Licensing agencies…; 3) assessing the implications of licensing on current software engineering practitioners and ameliorating any

negative implications." The proposal was to be brought back to the ACM Executive Council at its next meeting (June). It was now official. Something more permanent was soon to replace the Steering Committee (and it would have the power to work with state agencies much as Buckley had suggested early in 1993). But what that something would be remained to be decided—and it would be decided not by the Steering Committee as a whole but by Tripp and Frailey (after "consulting" with the Committee and others). Apparently, Tripp and Frailey were to have a free hand; the Steering Committee was already (more or less) dead.[33]

The ACM Executive Council, fifteen members present, met on Sunday at Washington's Renaissance Hotel from 8:30 AM to 4:10 PM (May 10, 1998). The part of the meeting devoted to Werth's report, the subsequent discussion, and the two motions was (judging by the minutes) about one thirteenth of the whole (30-40 minutes). A slightly larger part concerned Gotterbarn's report. The time set aside for the Steering Committee at this meeting (that is, for Frailey's report as well as Gotterbarn's) was more than a quarter of the whole, evidence of the importance the ACM assigned the Committee's work at this time.[34]

Gotterbarn began his presentation by explaining that the Code was a set of standards for a "sub-specialization" within the ACM and IEEE-CS, that the proposed draft (in one version or another) had been published on the web, in IEEE-CS's *Computer,* and the ACM's *Communications,* with a ballot to be returned, and that most provisions of the code had received a 95% approval in the balloting that followed. Gotterbarn asked the Executive Council to approve the Code more or less as it had approved the ACM Code six years before.

The discussion that followed did not go as Gotterbarn had expected. He soon learned that the Steering Committee had made its own recommendations concerning Version 4: Version 4 was "ready for review in a formal process". The process "should include balanced selection of representatives from software practitioners, educators, producers, and users" and "should follow the principles of due process such as those used in standards development."[35] Upon completion of the formal review, the Code would be "suitable for consideration by the appropriate bodies". The "appropriate committees of ACM and IEEE CS" should conduct the formal review. It was this motion, as Gotterbarn learned after some confusion, not his, that the ACM Executive Council officially had before it.[36] Gotterbarn also learned that Tripp was now IEEE-CS's President-Elect (that is, the President for 1999). "Mr. Standards" was now formidable indeed.

The rest of the discussion, though surprisingly peripheral, probably made the Executive Council hesitate to do anything definitive. One Council member, noting that the Code did not define "software engineer", asserted (perhaps in answer to Gotterbarn's description of software engineering as a "sub-specialty") that in Europe software engineering was "an integral part of Computer Science". Gotterbarn did not deny that, but reported that in Europe, or at least in Britain, computer science was itself considered part of engineering. That answer may have suggested the next question: If the ACM Code of Ethics is for computing, why have a separate code for software engineering? How is software engineering different? (The implicit premise of the question seems to be that a "sub-specialty" may, or may not, be different enough from the broader discipline to need its own code of ethics.) Gotterbarn responded that the Code "contains practical details of what makes SE an applied science". The response seemed to satisfy the Council on that question; the questions that followed were about the consistency between the Software Engineering Code and the ACM and IEEE codes. Had anyone checked for consistency? Gotterbarn could not say that anyone had but tried to dispose of any worry by pointing out that the Code's "purpose is to help the profession and not only ACM's membership

[the purpose of the ACM code]". The two codes did not apply to the same people. There was nonetheless agreement that a check for consistency should be done before the Executive Council approved the Code. Eventually, the Council unanimously asked COPE (the Committee on Professional Ethics) to "review the proposed [code] for compatibility with the ACM Code… and to determine the appropriate further process, if any, prior to adoption by the Council." At that time, COPE had three members. One was Robert Riser, one of Gotterbarn's colleagues at ETSU; a second was Keith Miller (a member of SEEPP's Executive Council); and the third, the chair, was Gotterbarn himself.[37] The Council had postponed approval but had refused to decide between Tripp and Gotterbarn. Like the IEEE-CS President, it had, in effect, told Gotterbarn to work things out with Tripp. That was not the worst outcome imaginable but—as Gotterbarn later said—"It was not a happy moment for me."[38]

11.3 Super-fast Standards process?

Gotterbarn now had no choice but to talk to Tripp. This he did soon after returning to Tennessee (probably early May 14).[39] The phone conversation began with Gotterbarn having no idea what to expect. He did not even know whether Tripp favored or opposed the Code. Yet, Gotterbarn's first impression of him was favorable. Though soft-spoken, Tripp clearly knew what he was doing and was business-like about doing it. He saw his job as supporting the production of the best—"most effective"—code possible. He had the same commitment to the Steering Committee's other task forces, Curriculum and Body of Knowledge. He wanted to make software engineering a profession. Following the IEEE Standards process was part of doing just that. The Code would not have much chance within IEEE-CS if anything but that process were followed. Engineers do not like to adopt Standards ad hoc. Tripp did not, however, expect the adoption process to take years. He had overseen the adoption of many Standards. He saw no reason why the whole process for Version 4, including approval by the IEEE-CS Board of Governors, could not be complete by autumn. The crucial step was getting the ballots out. Once they were out, the process would have a schedule—and automatically produce "closure" by a fixed date. Of course, to get the ballots out, "we" (Gotterbarn and Tripp) would have to develop a balanced list of reviewers, people whom both IEEE-CS and ACM members would respect as individuals and who, as a group, constituted a fair representation of the relevant constituencies. But Tripp had a staff experienced in developing balanced review groups, lists of potential reviewers, and so on. If Gotterbarn would prepare a preliminary list, including everyone he thought might be appropriate, Tripp would add whatever was missing. Tripp wanted the invitations to participate in the review group out by the end of May.[40]

By the time Gotterbarn hung up, he felt relieved and tired. He felt relieved because Tripp had seemed someone he could work with—competent, calm, with his heart in the right place—and because Tripp had an experienced staff to do much of the work. The nearest Gotterbarn had come to that sort of support was a student or two. Gotterbarn nonetheless felt tired after the conversation because he had not planned to devote his summer to getting the Code through the IEEE Standards process. He had promised to write several papers, had an NSF grant proposal to prepare, and was teaching summer school. He felt tired too because he doubted that anything as complex as the process Tripp described could be carried through as quickly as Tripp said it could.[41] Everything in Gotterbarn's experience told him to double or triple the time estimates on a project like that. But Gotterbarn had no alternative. If the Code were ever to be adopted, he had

to do as Tripp said. Gotterbarn quickly phoned Carver (IEEE-CS President) that he was working with Tripp.

A few minutes later, Tripp emailed Gotterbarn a one-page draft of "guidelines" and a draft ballot (with "15 July 1998" as the deadline for the end of balloting).[42] Later that day, Tripp also sent a draft "Invitation to Ballot" (with "1 June 1998" as the date to send it out). The arrival of these documents seemed a good omen. The next morning Gotterbarn wrote "Leonard",

> Boy do I like your efficiency!" The process you outlined looks fine. I have attached the three files you sent to me with some suggestions for modification. Let me know what you think.
>
> I am still developing a list of ballot invitees. I should have it in you[r] hands by Monday [May 18].[43]

Beneath Gotterbarn's signature (the usual "Don") were "Comments on the three documents". He had used "the comment function of word [MSWord]" to insert specific comments (another innovation in software). Anything he wanted to delete was in brackets and additions in quotes. He thought it important to use the version number of the Code since some of those voting on this version may have voted on Version 3. All references to the Code should be capitalized. He had some questions about the list he was preparing: "Are we still going with the categories of interest being: User of software engineered product[;] Producer of software engineered product[;] General (both user and producer)?" He wondered how to identify "Professional Engineers" (that is, licensed engineers). He even wondered what the official name of the Steering Committee is. (At the end of the Invitation, Tripp had referred to the committee he chaired as the "Software Engineering as a Discipline Steering Committee" after getting the name right at the beginning.)[44]

Within a few days, Gotterbarn sent Tripp about 270 names (with email addresses) drawn from (as he said in a note to himself) "various computer societies, cross section [of] people involved with licensing, Consultants, Academics, Professional Engineers, Consumers 'shrink wrapped', Custom software [users], anyone who responded to code, people from insurance company, Hitachi, Croatia, peter neuman."[45]

A few days after that, on May 26 (four days ahead of schedule), Tripp emailed the official invitation to participate in the balloting. It came from "Leonard L. Tripp, Chair, Joint Steering Committee for the Establishment of Software Engineering as a Profession", went to more than two hundred recipients addressed as "Dear Colleague", and had as the sender's email address twoods@computer.org (evidence, to those who knew about such things, that Tripp had staff support at IEEE-CS's Washington headquarters). The letter began with an invitation from the "IEEE Computer Society and ACM" to participate in the "formal review of version 4 of the Software Engineering Code of Ethics and Professional Practice." (The mention of ACM, and the inclusion of prominent ACM names on the list, was, it seems, intended to make a separate ACM ballot of membership unnecessary.) The letter briefly explained how the Code fit into the larger effort, begun in "November of 1993", to establish software engineering as a profession, concluding with the significant statement that the SEEPP task force "has completed its work and recommends that the Code be reviewed by a panel of peers using a formal review process."[46] The Code (Tripp continued) "addresses both the responsibilities of the practicing professional and of the profession". "We" (Tripp, SEEPP, the Steering Committee, or perhaps all) believe that

the societies "representing the profession" should "provide all software engineers with an ethical direction and with a tool they can use to educate others—including their management—about the ethical responsibilities of the software engineer."[47]

Having set a high tone, Tripp explained what participating in the "balloting group" entailed. The balloting group "will have 20 days to review and respond to the draft code." Reviewing the Code would be demanding: "The draft code is 7 pages long." The twenty days would probably be between June 22 and July 15, 1998. If interested, "you" should fill out the form at http://ada.computer.org/Ballot/Index.htm by June 12. By filling out that form, "you certify that you understand the subject matter of the proposed document and are technically competent to cast a ballot" Filling out the form also means "you agree to respond in the specified period". Responding in time would be important. The ballot would not be valid unless "at least 60% of the ballots sent [out]" were returned in time. For that reason, "members of a balloting group had an obligation to respond." The next to last paragraph provided further administrative detail (and directed all questions to "Tracy Woods, Volunteer Services Coordinator, IEEE-CS"); the last paragraph simply thanked the reader for "your time and participation".

Mechler received this invitation just before 6 PM on May 26. The next day, about noon, he received a short but hopeful follow-up message from Gotterbarn (sent through the listserv):

> You have all probably received an "Invitation to Ballot". This is the last formal step in processing the code. I hope you will all volunteer to be part of the balloting group and vote. The entire process—setting up the balloting group—sending [out] ballots, responding to negative ballots, producing a formal recommendation from the steering committee—is scheduled for completion by September—1998!

Around the world, others involved in software engineering were receiving Tripp's message—and some were also receiving Gotterbarn's. But at IIT, only Burnstein received Tripp's message—and, apparently, no one received Gotterbarn's. At least, we have no record of Gotterbarn's follow-up—or, indeed, of any other messages sent through the listserv in 1998.[48] This is odd because Gotterbarn and I were still corresponding all through 1998. Indeed, the day after he sent his follow-up message (May 28), I thanked him for "written comments" on an article he was "writing" for CSEP's newsletter *Perspectives*. Gotterbarn did not actually have time to write what I wanted. So, I interviewed him by phone, wrote up the interview as an essay of about 1500 words, and sent it to him for corrections, additions, and approval. (We had discussed the draft by phone, but he had also sent written comments.)[49] My note continued: "Some of the information [in the article] may also be useful later—if I ever get around to doing a book on professional codes, with your experience as the central example—an enormous success if the ACM and IEEE ever approve the code." I then promised to keep him informed concerning plans for what became this book "not least because I would want to do some interviews to fill in the many holes in the process as I know it—and to provide a useful step-by-step handbook of how to do it [write a code of ethics]." I cannot tell whether this letter was faxed or carried to its destination on a human shoulder. What seems probable is that I did not use email. Though the header includes an email address for Gotterbarn, it is an address that, by May 28, 1998, was almost a year out of date (Gotterbarn@ACCESS.ETSU.EDU). Communication between SEEPP and IIT (though not between Gotterbarn and me) had broken down—again.

11.4 The Vote

The breakdown of communication between SEEPP and IIT is one indication that selecting the balloting group may not have gone quite as planned. There were others. On May 27, Laurie Werth responded to Gotterbarn's email: "I leave town on June 6 and return on Aug 5—I fear that this will disqualify me unless I could vote before I leave. Is this possible?" We have this email because it survived in Mechler's files.[50] It did not survive in Gotterbarn's even through Werth sent it through the listserv. There is an explanation. In response to the writing of this book, Gotterbarn asked the ETSU system for a copy of his emails. What it did instead was to copy only the headers for 1998, deleting as it did all the content of the corresponding emails.[51]

Mechler apparently thought that the pool of potential voters should be as large as possible. He therefore forwarded the letter Tripp had sent him to anyone he thought should vote. Melcher's cover memo would say (something like):

> Received the invitation to ballot below and thought you may be interested. Yesterday I was on the web and applied. It looks like any one who visits the site or goes to it through IEEE CS site can ballot. Here are the web sites for the committee and the code if you want to look ahead of time.[52]

Was Mechler alone in sharing his invitation? How many voters entered the lists in this way? We do not know. What we do know is that such volunteers did not invalidate the process. The crucial step in achieving a balanced review panel was the final assembly of the group. The IEEE had standards for assuring an appropriate balance. If too many volunteered from any one group, only some of the volunteers would be given a ballot. If too few volunteered from any group, others from that group would be sought.[53]

When Mechler next wrote Gotterbarn, it was not to brag about how many potential voters he had signed up but to report that he "had [on June 29] tried to set confirmation on the list and got this response ['you are not subscribed to the PRFCMP-L list']." Observing that he seemed "not to be set", he asked whether he had missed anything. A few hours later, Gotterbarn informed Mechler that he had indeed missed something. The "[balloting] process is underway." The voting role had 192 names on it, of which Mechler's was one. "You will be getting email shortly about the web site we are setting up for the balloting." Gotterbarn then got around to the problem that had provoked Mechler's email. He had just "checked the prfcmp-l list and your name is listed as mechler_edmund@ADTRANZNA.COM." So, Gotterbarn continued, the listserv would not recognize "you" since "you" were using a different email address.[54] The next morning, Mechler thanked Gotterbarn "for the info" and explained how he came to ask the technical question Gotterbarn had answered the day before: "I never look at my e-mail [address] because everywhere else I have been, changed jobs three months ago, I had an internal and external e-mail address." But, where he was working now, he "found that I have another one for an unknown reason and during a change a couple of weeks ago the other was used when I sen[t]." He was just then "getting it fixed". He apologized for any trouble he had caused Gotterbarn, sketched what "set confirmation" meant (in response to a question from Gotterbarn), expressed satisfaction that the balloting was beginning, and concluded with details of how his IEEE-CS email address forwarded mail to the address he was actually using.[55]

The next morning (July 1), Mechler received Tripp's second "Dear Colleague" letter. This one informed him that he had been "selected to participate in the formal review" of the Code ("based on your response to our original Invitation to Ballot email in May"). Then, after taking a paragraph to repeat the history of the Steering Committee, Tripp instructed Mechler to go to a certain IEEE-CS web site where he would find three items: 1. Ballot Guidelines; 2. the Code; and 3. the Ballot Form. Tripp suggested taking "a moment to carefully read the Guidelines before proceeding to read the actual [Code]", referred all questions to "Tracy Woods at IEEE-CS", and stated that the deadline for return of the ballot was "Monday, 20 July 1998". The last two paragraphs were identical to those in his letter of invitation.[56] Tripp did not say it, but those who had kept his first letter could see that balloting was not quite on schedule. Tripp had said that it would (probably) take place between "June 22 and July 15, 1998". It was now scheduled to end five days beyond the bracketed dates.

On July 10, Woods sent out a "reminder" that the "deadline to respond to the Code of Ethics is Monday, 20 July", stressed the importance of getting a response rate of "at least 60%", and then pointedly reported that (half way through the twenty days of voting) "we have only reached 22%". She urged those with a ballot to "respond even if your vote is an abstention". Below her note she had helpfully placed Tripp's July 1 letter giving instructions on how to vote.

Early in the afternoon of July 20, the last day of voting, Mechler emailed Woods, "Did we make it???" Just after 8 AM the next morning, she replied, "Yes, as of Monday morning we had a 66% response rate. More came in throughout the day and overnight." The final response was 77% (148 out of 192 possible).[57] The vote was valid.

The votes now had to be counted, comments extracted, and a response devised for each comment. A computer at IEEE-CS automatically took the ballots off the web page, counted them, and placed each (including all comments) on a six-column spreadsheet (the first column for the number of the ballot, the second for date and time received, the third for the voter's name, the fourth for the vote, the fifth for "comments", and the sixth for "additional comments"). Version 4 had done well: 77% (113 of the 148) voted Affirmative, with 45 of those offering an optional comment. Only one percent (2) had voted Abstain. The rest 23% (33) were Negative.

Had Version 4 done as well as Version 3? Clearly, if we compare only raw percentages, the answer must be no. Support had dropped from 95% to 77%. But there are at least three reasons to discount such a comparison. First, those voting on Version 4 were not the same as those voting on Version 3. They may not even have been the same sort of people. Those voting on Version 3 consisted largely of IEEE-CS and ACM members who received the relevant society's general publication, read an article on ethics, and (with no other encouragement) filled out an eighty question survey. They paid $100 year to be a member of one (or both) of those societies. Though Gotterbarn had attempted to bring in some groups (primarily, corporate officers) who might not fit this (uncommon) profile, he had not systematically aimed at "balance". Tripp had. The overlap between those voting on Version 3 and those voting on Version 4 could not be more than 27%—and might be as little as 7%.[58] Those voting now may well have included many who, though willing to vote yes or no, would not have taken the time to fill out the elaborate survey the vote on Version 3 required. Those voting now may also have included many who voted out of devotion to the IEEE's Standards process, that is, people who voted because they received a specific emailed invitation. It certainly included two who, though they could not properly have voted on Version 3, happily voted on Version 4, Gotterbarn and Miller.

That is one reason to think the 77% favorable vote on Version 4 does not represent a loss of support when compared to the 95% vote on Version 3. Another is that there was no overall vote on Version 3; only on individual clauses. For all we know, many people who voted against even one clause would, if asked about the Code as a whole, have voted no. There is nothing irrational, or even unusual, about one clause in a complex document being a "deal breaker". Equally possible is that many who voted against many clauses might nonetheless, if asked, have voted for Version 3 as a whole—on the principle that, whatever its defects, Version 3 was much better than nothing.

A third reason not to treat the vote on Version 4 as comparable to that on Version 3 is that the voting procedure was different. Those voting on Version 4 knew (or, at least, should have known) that their vote and comments would be public (that is, posted on the IEEE-CS web site for all to see).[59] Those voting against the Code had to explain their vote (in the comments section)—or the negative vote would count as an abstention. They could not just complain; they had to propose an alternative (if only deletion of the offending language). They knew (or, at least, should have known) that if SEEPP accepted their alternative, their negative vote would automatically become positive. They would get another opportunity to vote only if SEEPP made some change in response to the negative comment but not exactly the change suggested—or only offered reasons not to make any change. Voting negative on Version 4 was part of a complex public conversation aiming at consensus, not a mere survey response (as the vote on Version 3 had been). Voters might well vote differently in environments so different.

While those three reasons seem to make a strong presumptive case for not treating the vote on Version 3 as comparable to the vote on Version 4, there is some reason to think that the two votes might nonetheless be comparable. In a few cases we know how a voter would have voted on both versions. Six people—Burnstein, Little, Langford, LaRue, Maner, and Mechler—voted in favor both times (or, in the case of Version 3, seem to endorse the code in such a way as to tell us that, if they had been asked, they would have explicitly endorsed the whole); and, of course, two more—Gotterbarn and Miller—would have done so had they thought to vote on Version 3. More interesting are the four—Berleur, Black, Harding, and Sigut—we know to have voted no on Version 4 and to have expressed a strongly negative opinion concerning Version 3.[60] All four wrote extensive comments as part of the balloting on Version 4; all complained of earlier suggestions not taken; and none complained about anything added between Version 3 and 4. They seem to have voted against Version 4 for the same reasons they would have voted against Version 3. Most interesting of all is the remaining category: those known to have voted yes on one version and no on the other. That category seems to be empty. We are then entitled to conclude that voting on the two versions was comparable enough to preserve consistency in response. What might comparing the two results tell us?

It will not tell us that Version 4 had lost 20% of the support Version 3 had. We cannot know about that without doing a comparison of all the people in the two voting pools—along whatever are the right axes for predicting how they would vote. I do not know how to do that. What comparison of the two votes does tell us, if it tells us anything, is that all the work that turned Version 3 into Version 4 probably had little effect (positive or negative) on the support Version 4 received. The category of yes-on-one-and-no-on-the-other is empty. Had Version 3 been put to the same test as Version 4, Version 3 must (assuming the category of vote changers really is empty) have received a favorable rating of about 77%. Is the same true of Version 2.1—and even of Version 1.0? That is certainly possible on the evidence we have. Are we to conclude,

then, that had Gotterbarn teamed up with Tripp in November 1996 rather than in May 1998, the whole process could have ended eighteen months earlier?

Mechler once told me that he thought Gotterbarn's various versions of the Code amounted to "moving deck chairs around on the *Queen Mary*". [61] He did not mean that the succession of versions was as futile as the proverbial "moving deck chairs around on the [iceberg-gashed] *Titanic"*. On a ship likely to reach port in a few days, the location of deck chairs matters—but, in large part at least, only to the passengers. Their location will not speed the ship on its way or delay its arrival by even a minute. In writing codes, as in arranging chairs on deck, there is considerable "free play"—affecting the look, feel, and content of the code, but not its likelihood of ultimate approval. At least one other member of SEEPP/E holds a similar view. According to Jayaram, "He [Mechler] kept us going until we drafted the code (Version 1). Everything after that was minor in comparison."[62]

Questions about what might have been are, of course, notoriously hard to answer, especially in human affairs, and I do not claim to know what would have happened had Tripp and Gotterbarn teamed up eighteen months earlier. I do, however, think it worth noting a little more evidence that Gotterbarn's successive revisions may not (overall) have improved the Code's chance of adoption (however much they may have improved the Code). A number of the negative votes cite as one reason (or *the* reason) for the negative some way in which Version 4 differs from one or more of its predecessors. So, for example, George Samaras (#25) objects to "economic disadvantage" in Clause 1.07. "If the software is too expensive," he reasoned, "it will not be used/will not sell." The issue is economic, not ethical. 1.07 made its first appearance in Version 2.1 (as the ever-controversial "diversity" clause 2.05). Or, consider the next negative vote on Version 4: William S. Junk (#27) voted no in part because "1.03 ['Approve software only if they have a well-founded belief that it is safe, meets specifications, has passed appropriate tests, and does not diminish quality of life or harm the environment'] does not seem to recognize that there are legitimate applications of software technology that may have negative consequences on the quality of life of some individuals or result in some harm to the environment." Clause 1.03 is the successor of Version 1's 2.02 ("Approve software only if they have a well-documented belief that it is safe, meets specifications, and has passed all appropriate tests"). The chief difference between the two (or, at least, the difference relevant here) is the addition of consideration of "quality of life" and "harm to the environment", the very phrases that concern Junk. Junk argues that there are legitimate projects for software engineers, for example, a "smart weapons system", that cannot avoid reducing someone's quality of life (the target's) or harming the environment (as collateral damage). Junk is nonetheless willing to change his vote if (along with changes in some other provisions), the word "substantially" were inserted before "diminish" and "harm".[63]

We are, however, not entitled to conclude from Junk's objection to 1.03 that the executive committee made a mistake by expanding the coverage of old 2.02 or that it could have increased the total vote for Version 4 by anticipating Junk's amendment. It is at least possible that some positive votes depend on 1.03 not being weakened in the way Junk suggested. Having pointed out that possibility, we can say no more. While the executive committee undoubtedly had a certain amount of freedom in what it did with 1.03, it did not know—and could not have known—how much freedom it had. Indeed, even now we do not know that.

11.5 Can this really be the end?

Balloting closed on July 20. On July 22, Woods sent Gotterbarn the votes and comments. Gotterbarn suggested removing some personal information from two entries. On July 24, when Woods posted the official (slightly edited) spreadsheet on the IEEE-CS website, Gotterbarn was already at work on his responses. He had to respond by accepting each proposed change, by making some lesser change, or by explaining why he (or, rather SEEPP) had left the provision as it was. Gotterbarn had about three weeks according to Tripp's schedule—and only a few days more than that till the start of the Fall Term.

Gotterbarn had at least two reasons not to revise the Code again. First, ninety-three voters had put something in the comment section (with twenty of these also making "additional comments"). Once these comments were organized into distinct points (with one or more people making the same point), each required a distinct response—as well, perhaps, as a revision in the Code.[64] Responding would be a big job, responding in a few weeks, bigger yet; and carefully reconsidering every provision someone challenged, a very big job indeed.

The second reason Gotterbarn had not had to revise the Code again was that Version 4 had passed the formal review process already. According to Tripp, 60% was the minimum to pass.[65] All Gotterbarn could hope to do now was raise the positive percentage above the already comfortable 77%. There was *no* guarantee that any further revision would in fact do that. Votes gained might be balanced or over-balanced by votes lost. Yet, working with his executive committee in a way by now familiar, Gotterbarn did much more than the minimum the procedures required.[66] By August 6, when he met with Miller and Rogerson at the "Tangled Web" conference at Dartmouth College, he had a document sufficiently different from Version 4 to be called "Version 5.0" and a draft of 185 "responses", explaining what had (and had not) been done in response to various comments.[67] The three found time during the conference to review and revise both documents.[68] Then it was off to Boston to deliver a paper on August 10 at the World Congress of Philosophy. Gotterbarn sent Version 5.0 and the comments to Tripp (as attachments) as soon as he returned to Johnson City. Tripp acknowledged receipt about 4 PM, August 12 (Wednesday), asking Gotterbarn to "resend the second file" (the comments) because he could read only the first file (with "version 5" in it). Guessing that the problem was the second file's name, Tripp recommended that Gotterbarn shorten the file name to eight characters or less before trying to resend.[69] Tripp concluded this email with a reminder: "Next step is recirculate the revised document and the comment resolution to the balloting group. The target is to submit the document and the comment resolution to the leadership of the two societies for acceptance."

On August 31, the 185 responses were published in the sixth column of the spreadsheet reporting results of balloting on Version 4.[70] On September 9 (fifty days after the close of voting), Tripp emailed everyone who had participated in the balloting his third "Dear Colleague" letter.[71] It not only reported the results of the first round of balloting but the results of work done since. Thirteen negative votes had been "resolved" (with twenty still "unresolved"), raising the affirmative from 77.4% to 86.3%.[72] It was now time for a "recirculation ballot". The purpose of this ballot was to "confirm that comments during the initial ballot have been dealt with in an acceptable manner" and also to "increase consensus on the Code of Ethics document".

Tripp then directed voters to the appropriate IEEE-CS website, asking them "to review the Code of Ethics (Version 5.0) and to complete the Recirculation Ballot."[73] Those who did not

wish to change their vote need do nothing. Negatives would remain negative; positives, positive. (Silence did not constitute consent, except for those who had already consented.) There were only four reasons to change a vote: First, those who thought all their objections had been "addressed" could change their negative vote to affirmative. Second, those who thought "one or more" of their objections had not been addressed could "reiterate [their] negative vote". Third, those who wished to "indicate [their] support of objections of another person which have not been resolved" could change an affirmative vote to a negative. Fourth, those who thought "changes made to the document have introduced substantive problems that did not previously exist" could change their affirmative vote to negative. The deadline for responding was September 16 (one week from the date this email was sent out). Tripp ended the letter inviting "comments on the balloting process", asking that "the questionnaire on the same Web page" be completed, and directing any "technical or procedural questions" to his own (Seattle) phone number. [74]

On September 23, Woods issued the final tally. She listed thirty-two voters as originally negative.[75] Of these, only twelve had voted again. Half of these had changed their vote to affirmative; half (including Junk and Sigut) had voted negative again.[76] Did any affirmative votes change to negative? The answer *seems* to be no, but even Gotterbarn does not *know*. What he reported to the listserv on October 13, 1998, is all he recalls today: "I do not have the official final word, but 6 of the unresolved negative votes changed to positive so I think the final analysis is 132/146 affirmative (90%). That is a passing grade :-)" He then added that the Steering Committee would be "presenting the Code to the ACM Council….this month and to the IEEE next month….It may be a great Thanksgiving."[77] Seven days later (October 20), he wrote the listserv again: "Yesterday the ACM Council met in Vancouver and approved the Software Engineering Code of Ethics and Professional Practice (5.1) as their approved Code for the practice and teaching of Software Engineering!!!!!!!!!!" The vote was unanimous. There had not even been one abstention. He then added that the "FINAL version" was "on the seeri web site….I am exhausted…….more later."[78]

This email started another listserv party (the first having been almost a year earlier). Laurie Werth replied within an hour: "congratulations!!" A few minutes later, Norman did the same: "This is great news. Thank you Don for all the hard work and encouragement." Werth then had second thoughts: "Yes, but is it Y2k-compliant?" Langford chimed in from England (adding below Norman's message): "Hear, hear…hard taskmaster as he undoubtedly was, Don nursed this project to success against considerable odds, and thoroughly deserves congratulations from us (and everyone else!)". The last to join the party (and perhaps the one who killed it with a touch of reality) was Mechler:

> Congratulations Don and condolences for the IEEE meeting.
> Congratulations to the whole team.
> Has it been determined where the code will reside?
> Maybe success has gone to my head but is there a next step we should consider?[79]

Amr El-Kadi emailed Gotterbarn directly two days later (October 22, 1998): "I[t] was exceptional news that we have reached this stage with Code, thanks to your leadership. Ed noted there were problems with IEEE, what happened there"?[80]

Too exhausted and exhilarated to write much, Gotterbarn had omitted some interesting details about what happened at the ACM Council meeting on October 19. He had taken time during his presentation in Vancouver to respond to the three issues the Council had raised in Washington in May. First, COPE had compared the ACM Code with the Software Engineering Code and found no incompatibility. Second, the two codes were not in competition. Those covered by the two codes overlapped but were not coextensive. Not all ACM members are software engineers; not all software engineers are ACM members. Nor was there anything odd about one person being subject to two codes of ethics. A Professional Engineer, for example, is subject to the code of ethics of both the National Society of Professional Engineers and that of his branch of engineering (such as the code of ethics of American Society of Mechanical Engineers). Third, COPE had recommended a four-stage approval process (which seems much more orderly as described here than it did as it developed):

> A. Develop and revise the Code based on extensive review [completed "July 1997"].
> B. Have the Code reviewed by the Joint ACM/IEEE-CS Task Force [this had included the Fall 1997 publication of Version 3 and ACM/IEEE-CS membership voting on it and resulted in the revisions that produced Version 4].
> C. Use an existing and tested formal review process used by the IEEE for approving international technical standards [Version 4 had easily passed that test but, as a result of suggestions made during the balloting, had been revised twice, generating Version 5.1].
> D. Formal adoption of [Version 5.1] by the ACM and Computer Society…as their approved code for the practice and teaching of software engineering.[81]

Though the ACM Council unanimously passed a motion appropriate to stage D (the last), it did not approve precisely the motion Gotterbarn expected. The Council approved the Code on condition an ACM "grammarian" examine it. At least one member of the Council was unhappy with some split infinitives.[82]

On November 20 (still, technically, a month before the last day of autumn), the IEEE-CS Board of Governors also voted to approve the Code (Version 5.1). But it too had a condition: its approval was subject to IEEE's lawyer examining the document and determining that approval would not cause IEEE-CS any legal problems.[83] On December 10, the lawyer, Sandra Pfau, answered the Board's concerns in a one-page memo to Anne Marie Kelly, Director, Volunteer Services, IEEE-CS (Woods' superior). Pfau stressed that standard setting is always legally risky, offering this chilling footnote:

> A recent example of the extensive liability associated with association standard setting is the case of National Spa & Pool Institute of Alexandria, Virginia. Due to a $6.6 million judgment against the Institute related to a man's life-changing injuries in a 1991 diving board incident, the institute has had to declare bankruptcy to keep operating while it appeals.

Voluntary standards are, however, "less problematic in the legal sense" than certification programs or "product standards". For this reason, "the language included in the code of ethics is perhaps not as significant as the manner in which the code is disseminated and publicized." The IEEE-CS should "make it clear when the code is published that it is a voluntary code." More

important, when the IEEE-CS endorses the Code, it should "make it clear that the endorsement goes to the use of the code, not to individuals or companies that claim to abide by the code." The distinction Pfau had in mind seems to be between endorsing the Code and certifying (or otherwise endorsing) specific practitioners or products. That is indeed an important distinction, but not one that divides "voluntary" from "involuntary" codes. Her next sentence (and the whole paragraph immediately below it) collapses the one distinction into the other. While the IEEE-CS "can say that it believes that following the code is a good thing, it is not a good idea…to say or imply that those who follow or have adopted the code are better software engineers than those that have not adopted the code." She was, of course, right that merely (formally) adopting the Code would not make one a better software engineer. But, unless everyone involved in writing the Code was badly mistaken, she must be wrong that *following* the Code does not make one a better software engineer. Of course, it does. That is why the IEEE-CS would—and should—say that "it is a good thing" to follow the Code.

The IEEE-CS Board of Governors bravely took this memo as favoring approval. The lawyer had, however, noted one "typo". Having corrected it, along with the faults the ACM grammarian had identified, Gotterbarn had before him Version 5.2 of the Software Engineering Code of Ethics and Professional Conduct, approved both by the ACM and the IEEE-CS (11.Appendix). It is hard to know exactly when this happened. But there is a memo that Gotterbarn sent Tripp on January 18, 1999, in which Gotterbarn both admitted his pleasure "that the Code has passed legal muster" and informed Tripp that "clause 6.13" had been corrected (as Pfau had suggested).[84] So, the process of approval was complete no later than January 18—and perhaps a few days before that, just under a month later than Tripp had estimated in May and just over five years after Gotterbarn had met with Barbacci and Melford in Pittsburgh to organize SEEPP. When the great moment came, it was an anticlimax.

11. Appendix**:**

SOFTWARE ENGINEERING CODE OF ETHICS AND PROFESSIONAL PRACTICE

(Version 5.2) as recommended by the
IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices and
Jointly approved by the ACM and the IEEE-CS as the standard for teaching and practicing software engineering.

Short Version

PREAMBLE
The short version of the code summarizes aspirations at a high level of abstraction. The clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

1 PUBLIC - Software engineers shall act consistently with the public interest.

2 CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest.

3 PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4 JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.

5 MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6 PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7 COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.

8 SELF - Software engineers shall participate in lifelong learning regarding the practice of their

profession and shall promote an ethical approach to the practice of the profession.

SOFTWARE ENGINEERING CODE OF ETHICS AND PROFESSIONAL PRACTICE

IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices

Full Version

PREAMBLE

Computers have a central and growing role in commerce, industry, government, medicine, education, entertainment and society at large. Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance and testing of software systems. Because of their roles in developing software systems, software engineers have significant opportunities to do good or cause harm, to enable others to do good or cause harm, or to influence others to do good or cause harm. To ensure, as much as possible, that their efforts will be used for good, software engineers must commit themselves to making software engineering a beneficial and respected profession. In accordance with that commitment, software engineers shall adhere to the following Code of Ethics and Professional Practice.

The Code contains eight Principles related to the behavior of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession. The Principles identify the ethically responsible relationships in which individuals, groups, and organizations participate and the primary obligations within these relationships. The Clauses of each Principle are illustrations of some of the obligations included in these relationships. These obligations are founded in the software engineer's humanity, in special care owed to people affected by the work of software engineers, and in the unique elements of the practice of software engineering. The Code prescribes these as obligations of anyone claiming to be or aspiring to be a software engineer.

It is not intended that the individual parts of the Code be used in isolation to justify errors of omission or commission. The list of Principles and Clauses is not exhaustive. The Clauses should not be read as separating the acceptable from the unacceptable in professional conduct in all practical situations. The Code is not a simple ethical algorithm that generates ethical decisions. In some situations, standards may be in tension with each other or with standards from other sources. These situations require the software engineer to use ethical judgment to act in a manner which is most consistent with the spirit of the Code of Ethics and Professional Practice, given the circumstances.

Ethical tensions can best be addressed by thoughtful consideration of fundamental principles, rather than blind reliance on detailed regulations. These Principles should influence software engineers to consider broadly who is affected by their work; to examine if they and their colleagues are treating other human beings with due respect; to consider how the public, if reasonably well informed, would view their decisions; to analyze how the least empowered will be

affected by their decisions; and to consider whether their acts would be judged worthy of the ideal professional working as a software engineer. In all these judgments concern for the health, safety and welfare of the public is primary; that is, the "Public Interest" is central to this Code.

The dynamic and demanding context of software engineering requires a code that is adaptable and relevant to new situations as they occur. However, even in this generality, the Code provides support for software engineers and managers of software engineers who need to take positive action in a specific case by documenting the ethical stance of the profession. The Code provides an ethical foundation to which individuals within teams and the team as a whole can appeal. The Code helps to define those actions that are ethically improper to request of a software engineer or teams of software engineers.

The Code is not simply for adjudicating the nature of questionable acts; it also has an important educational function. As this Code expresses the consensus of the profession on ethical issues, it is a means to educate both the public and aspiring professionals about the ethical obligations of all software engineers.

PRINCIPLES

Principle 1 PUBLIC Software engineers shall act consistently with the public interest. In particular, software engineers shall, as appropriate:

1.01. Accept full responsibility for their own work.
1.02. Moderate the interests of the software engineer, the employer, the client and the users with the public good.
1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.
1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.
1.05. Cooperate in efforts to address matters of grave public concern caused by software, its installation, maintenance, support or documentation.
1.06. Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.
1.07. Consider issues of physical disabilities, allocation of resources, economic disadvantage and other factors that can diminish access to the benefits of software.
1.08. Be encouraged to volunteer professional skills to good causes and to contribute to public education concerning the discipline.

Principle 2 CLIENT AND EMPLOYER Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest. In particular, software engineers shall, as appropriate:

2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.
2.02. Not knowingly use software that is obtained or retained either illegally or unethically.
2.03. Use the property of a client or employer only in ways properly authorized, and with the client's or employer's knowledge and consent.
2.04. Ensure that any document upon which they rely has been approved, when required, by someone authorized to approve it.
2.05. Keep private any confidential information gained in their professional work, where such confidentiality is consistent with the public interest and consistent with the law.
2.06. Identify, document, collect evidence and report to the client or the employer promptly if, in their opinion, a project is likely to fail, to prove too expensive, to violate intellectual property law, or otherwise to be problematic.
2.07. Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.
2.08. Accept no outside work detrimental to the work they perform for their primary employer.
2.09. Promote no interest adverse to their employer or client, unless a higher ethical concern is being compromised; in that case, inform the employer or another appropriate authority of the ethical concern.

Principle 3 PRODUCT Software engineers shall ensure that their products and related modifications meet the highest professional standards possible. In particular, software engineers shall, as appropriate:

3.01. Strive for high quality, acceptable cost, and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.
3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.
3.03. Identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects.
3.04. Ensure that they are qualified for any project on which they work or propose to work, by an appropriate combination of education, training, and experience,.
3.05. Ensure that an appropriate method is used for any project on which they work or propose to work.
3.06. Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified.
3.07. Strive to fully understand the specifications for software on which they work.
3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements and have the appropriate approvals.
3.09. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates.
3.10. Ensure adequate testing, debugging, and review of software and related documents on

which they work.
3.11. Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.
3.12. Work to develop software and related documents that respect the privacy of those who will be affected by that software.
3.13. Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.
3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrences.
3.15 Treat all forms of software maintenance with the same professionalism as new development.

Principle 4 JUDGMENT Software engineers shall maintain integrity and independence in their professional judgment. In particular, software engineers shall, as appropriate:

4.01. Temper all technical judgments by the need to support and maintain human values.
4.02 Only endorse documents either prepared under their supervision or within their areas of competence and with which they are in agreement.
4.03. Maintain professional objectivity with respect to any software or related documents they are asked to evaluate.
4.04. Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.
4.05. Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.
4.06. Refuse to participate, as members or advisors, in a private, governmental or professional body concerned with software related issues, in which they, their employers or their clients have undisclosed potential conflicts of interest.

Principle 5 MANAGEMENT Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance. In particular, those managing or leading software engineers shall, as appropriate:

5.01 Ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk.
5.02. Ensure that software engineers are informed of standards before being held to them.
5.03. Ensure that software engineers know the employer's policies and procedures for protecting passwords, files and information that is confidential to the employer or confidential to others.
5.04. Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.
5.05. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work, and provide an uncertainty assessment of these estimates.
5.06. Attract potential software engineers only by full and accurate description of the conditions of employment.

5.07. Offer fair and just remuneration.
5.08. Not unjustly prevent someone from taking a position for which that person is suitably qualified.
5.09. Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed.
5.10. Provide for due process in hearing charges of violation of an employer's policy or of this Code.
5.11. Not ask a software engineer to do anything inconsistent with this Code.
5.12. Not punish anyone for expressing ethical concerns about a project.

Principle 6 PROFESSION Software engineers shall advance the integrity and reputation of the profession consistent with the public interest. In particular, software engineers shall, as appropriate:

6.01. Help develop an organizational environment favorable to acting ethically.
6.02. Promote public knowledge of software engineering.
6.03. Extend software engineering knowledge by appropriate participation in professional organizations, meetings and publications.
6.04. Support, as members of a profession, other software engineers striving to follow this Code.
6.05. Not promote their own interest at the expense of the profession, client or employer.
6.06. Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest.
6.07. Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.
6.08. Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.
6.09. Ensure that clients, employers, and supervisors know of the software engineer's commitment to this Code of ethics, and the subsequent ramifications of such commitment.
6.10. Avoid associations with businesses and organizations which are in conflict with this code.
6.11. Recognize that violations of this Code are inconsistent with being a professional software engineer.
6.12. Express concerns to the people involved when significant violations of this Code are detected unless this is impossible, counter-productive, or dangerous.
6.13. Report significant violations of this Code to appropriate authorities when it is clear that consultation with people involved in these significant violations is impossible, counter-productive or dangerous.

Principle 7 COLLEAGUES Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, as appropriate:

7.01. Encourage colleagues to adhere to this Code.
7.02. Assist colleagues in professional development.

7.03. Credit fully the work of others and refrain from taking undue credit.
7.04. Review the work of others in an objective, candid, and properly-documented way.
7.05. Give a fair hearing to the opinions, concerns, or complaints of a colleague.
7.06. Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and security measures in general.

7.07. Not unfairly intervene in the career of any colleague; however, concern for the employer, the client or public interest may compel software engineers, in good faith, to question the competence of a colleague.
7.08. In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.

Principle 8 SELF Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession. In particular, software engineers shall continually endeavor to:

8.01. Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.
8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.
8.03. Improve their ability to produce accurate, informative, and well-written documentation.
8.04. Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.
8.05. Improve their knowledge of relevant standards and the law governing the software and related documents on which they work.
8.06 Improve their knowledge of this Code, its interpretation, and its application to their work.
8.07 Not give unfair treatment to anyone because of any irrelevant prejudices.
8.08. Not influence others to undertake any action that involves a breach of this Code.
8.09. Recognize that personal violations of this Code are inconsistent with being a professional software engineer.

This Code was developed by the IEEE-CS/ACM joint task force on Software Engineering Ethics and Professional Practices (SEEPP):
Executive Committee: Donald Gotterbarn (Chair),
Keith Miller and Simon Rogerson;
Members: Steve Barber, Peter Barnes, Ilene Burnstein, Michael Davis, Amr El-Kadi, N. Ben Fairweather, Milton Fulghum, N. Jayaram, Tom Jewett, Mark Kanko, Ernie Kallman, Duncan Langford, Joyce Currie Little, Ed Mechler, Manuel J. Norman, Douglas Phillips, Peter Ron Prinzivalli, Patrick Sullivan, John Weckert, Vivian Weil, S. Weisband and Laurie Honour Werth.

NOTES

[1] Of course, this "politically correct" and "American" language entered the Code when Gotterbarn was working in England under the watchful eye of that Scot-patriot Simon Rogerson. Gotterbarn was then trying to make the code "more international". It is remarkable how different from our intentions our achievements can be (or, at least, seem to others)!

[2] The Dutchman also mentioned (old) 5.06 ("Attract employees only by full and accurate description of the conditions of employment") in the same sentence, but apparently as an example of a standard already "enforced by local law". Gotterbarn\Survey Comments\ SLEGERS. Clause 5.05 had begun as 6.10 in Version 1; 5.06, as 6.12. The survey had given each a favorable rating just under 94%. Gotterbarn\Version 3\V3 Survey votes. Why had the two been deleted? Not all provisions receiving less than 95% had been deleted. Some, like 1.03 (the old 2.02), had received only 90% approval—and would continue to cause trouble. How did Gotterbarn decide what to revise and what to leave the same?

[3] Interview of Sigut, September 27, 2002. His education identifies him as an engineer (strictly so called): "I received a Diploma in Engineering in Mechanical Engineering in 1972 (roughly equivalent to an MS) from the Federal Institute of Technology (ETH for *E*idgenoessiche *T*echnische *H*ochschule). I began work as a structural engineer in 1972, using computers as part of the job. As time went on, writing software became an ever-bigger part of the job, until I came to think of myself as a software engineer. There was not then a career path leading to software engineering."

[4] Compare Sigut's comments in the July 20 vote (Gotterbarn\Version 5\RPLY):

> Second sentence of the third paragraph of the PREAMBLE to the Full Version says: The list of Principles and Clauses is not exhaustive, and should not be read as separating the acceptable from the unacceptable in professional conduct in all practical situations. When we analyse this sentence, we can say about "the list" that it contains a. Principles b. Clauses ... and that it ... c. is not exhaustive d. should not be read as separating ... We therefore have two distinctive parts of the list, which can be said to have two properties each. Let us look at the four possible combinations: 1. The list of Clauses is not exhaustive OK, the list of Clauses can of course be continued infinitely. 2. The list of Clauses should not be read as ... OK, Necessary condition, otherwise inflexible. 3. The list of Principles is not exhaustive… the list of Principles IS the base of the Code. Of course, we might later find omissions, but that can be adjusted when we get there. If we say NOW that the list of Principles is not exhaustive, we better sit down and think again. 4. The list of Principles should not be read as ... It is not the Principles, but it is their application - embodied e.g. in the Clauses - , which can separate the acceptable from the unacceptable. It is not necessary to state that something impossible does not apply. Considering the four combinations we can see, that while the properties of the Clauses are well defined by the sentence, we can NOT say the same about the properties of the Principles. I therefore propose the sentence to be changed to read: The list of Clauses is not exhaustive, and should not be read as separating the acceptable from the unacceptable in professional conduct in all practical situations.

[5] Clause 7.06 survived unchanged through Version 5.2.

[6] Sigut Archive: 1-8-98. Sigut responded to Version 4 on January 12. That response is a bit over a page long. Most of his suggestions are minor (including several typos to fix), but he does "again…plead for" omitting "Principles and" because "while I agree with 'the list of Clauses' not being exhaustive, I have strong doubts and misgivings about saying the same about the Principles." This time (apparently) Gotterbarn did not respond. The objectionable language survives in Version 5.2.

[7] This provision is, of course, another one in need of editing: "on which…they…propose" is ungrammatical. 3.02 should read (something like): "Ensure proper and achievable goals and objectives for any project on which they work *or propose to work*" (or perhaps "Ensure proper and achievable goals and objectives for any project *which they propose or* on which they work"). This unfortunate sentence survives in Version 5.2.

[8] Zweben's letter is lost. We have only the response, Gotterbarn\Version 4\STURESP (no date on letter but file date is 1-19-1998). Apparently, Zweben had suggested "balance" instead of "moderate" (nearly an exact synonym). Gotterbarn did not explain how this weaker provision (in either form) was consistent with moving Principle "Public" from third place to first, with the claim in the Preamble that the public good is "primary", or with various Principles requiring that serving the client, employer, and profession should be done "consistent with the public interest". This language survive into Version 5.2.

[9] Indeed, except for its date (January 19), its first paragraph, the three sentences of the second paragraph, and the salutation ("Dear Steering Committee for the Professionalization of Software Engineering"), it is identical to the one sent December 16.

[10] Gotterbarn\Version 4\REC2 (or the identical: Gotterbarn\Steering Committee\ Recommendation of 4\REC2).

[11] Gotterbarn\Version 4\REC2.

[12] Gotterbarn\Version 4\Progress. The letter nonetheless carries "19 January 1998" as its date (and that is also the file date). Gotterbarn\Version 4\REC2.

[13] Gotterbarn\Version 4\Progress.

[14] Cabrera seems to have resigned from the Committee at the same time he resigned as chair. There was, it seems, no replacement.

[15] During Spring 1998, Gotterbarn used the title "co-chair" to describe the chair or vice-chair of the Steering Committee, apparently slipping into the language of his own early days working as one of SEEPP's two "co-chairs".

[16] The site www.computer.org/tab/seprof still existed on March 16, 2003 (without any obvious updating from five years before). Everything at that site, except Version 4, dated from March, 1997, another sign that the Steering Committee had been disintegrating for almost a year.

[17] According to Gerald Engel (then co-chair of the Curriculum Task Force), things were not as bad as Gotterbarn thought. ABET's request concerned accreditation criteria (something much less detailed than a curriculum). The work of the Curriculum Task Force would be (more or less) independent of what ABET was doing. Engel admits that this is a point likely to be lost on everyone but an expert in accreditation and curriculum.

[18] Charles (Chuck) House was then Intel's Science Policy and Society Impact Director. Before joining Intel in 1995, he had been President of Spectron MicroSystems, a wholly-owned subsidiary of Dialogic, focusing on distance learning and "corner conference room" enhancements. He had long been active in the IEEE as well as in the ACM (at one time serving as an IEEE Vice-President for Publications). www.cpd.ogi.edu/egm/house.htm (3/23/2004). He did not respond to email, classic mail, or phone messages requesting an interview.

[19] Gotterbarn\Version 4\PRESIDAC. I attribute the letter to Gotterbarn even though all three members of SEEPP's executive committee are listed as signatories. While not much turns on the attribution, I feel more comfortable treating Gotterbarn as sole author both because the letter uses "I" much more often than "we" (though it uses both) and because some of the detail (for example, events at a conference) seem personal. While I am sure that Gotterbarn cleared the letter with Miller and Rogerson, there is in fact no record even of that in the archives.

[20] Gotterbarn did not say it, but he knew about these forthcoming publications because they were his, part of the strategy of dissemination to prevent suppression. See: "The Uniqueness of Software Errors and Their Impact on Global Policy", *Science and Engineering Ethics* 4 (July 1998): 351-356; "Not all Codes are Created Equal: The Software Engineering Code of Ethics, a success story", *Journal of Business Ethics* 22 (October 1999): 81-89); and "The Ethical Computer Practitioner—Licensing the Moral Community: a proactive approach", *SIGCSE Bulletin* 30 (June 1998): 8-10. "The Proceedings of the ACM Policy Conference in May" were printed in a special issue of *Computer and Society* (with Keith Miller editing it along with Tom Jewett). Gotterbarn's contribution was: "Raising the bar: a software engineering code of ethics and professional practice", *Computers and Society* 28 (June 1998): 26-28 (with Version 4 in the next section). Gotterbarn was then vice-chair (and "Information Director") of the Special Interest Group (SIG) sponsoring that journal. If there was an ethics network within ACM, Gotterbarn was now at its center.

[21] Gotterbarn\Steering Committee\Dennis.

[22] Would the ACM Executive Council also notice that the short version said it was not to be severed from the long version because the short version's "aspirations can become high sounding but empty" without the details in the long version? Did Gotterbarn now silently thank Fairweather for insisting that something be said in the introduction to the short version to prevent its being severed from the long?

[23] Gotterbarn Chapter10cmt (September 30, 2004).

[24] www.computer.org/csinfo/bios/tripp.htm (3/16/2004).

[25] "[One] of the things we were warned about using as a model very early by Elliot C at a steering committee meeting was going through this process which normally takes YEARS because of infighting by vendors and review processes." Email (Gotterbarn to Davis), March 17, 2004. Compare Ch. 3.6.

[26] Gotterbarn\Version 4\Tripp1. There is no evidence that Tripp acknowledged receipt of this report.

[27] Mechler\SEEPP98.

[28] Gotterbarn\Version 4\ACM2 (with letterhead) and Gotterbarn\Version 4\ACM1 (without letterhead or addressee).

[29] Apparently, Gotterbarn communicated with Frailey again, by phone or email, after April 8. The archive does not contain that email (or any other to or from Frailey) before the May 10 meeting (or after).

[30] Gotterbarn\Version 4\Ethics Report (fwd). Carver—and her ACM co-chair—were both replaced at the end of 1996. The new co-chairs were Gerald Engel (IEEE-CS) and Rich LeBlanc (ACM). They served until the work of the task force was completed in 1998. www.acm.org/serving/se/Rpt9811.htm (3/23/2004).

[31] Item 3.2. Gotterbarn\Version 5\acmsupport\POL98MIN.

[32] For Frailey's overheads for that report, see www.acm.org/serving/se/min990128 (4/11/2004), attachment entitled "History of SWECC". Frailey titled this set of overheads the Steering Committtee's "Final Report". The overheads report two recommendations. Beside recommending "formal review, adoption, and approval processes be followed", it also recommended "continuing the code of ethics task force because of ongoing needs in professional practices area."

[33] For details of the report they delivered in October 1998, see 12.1 (and Gotterbarn\Version 5\ACMSUPPORT\ Report).

[34] The Steering Committee's share is even larger if we include item 3.6—an update on the integration of the CSAB with ABET. (There is a reference under that item to the Steering Committee's task force on Curriculum.) The motion in support of the CSAB-ABET Memo of Agreement passed without opposition but with three abstentions. For those wondering what "CSAB" stands for, the answer, according to Article I of its Constitution (October 15, 2000), is "CSAB Inc." (not, as widely supposed, "Computing Sciences Accreditation Board").

[35] The minutes attribute the substance of this paragraph to the "TF" (presumably, the "task force", not "Tripp and Frailey"). Gotterbarn thinks the minutes should have said "Steering Committee". Email (Gotterbarn to Davis), March 17, 2004. There are, however, at least two explanations of "TF" that would not reduce it to a mere recording error. One is that the Council members were as unclear about the distinction between the task force and Steering Committee as were many of us who participated in SEEPP's work. The other possibility (perhaps the more likely) is that the recording secretary was simply echoing the language of the Steering Committee motion—and that (like the letter Tripp was to write two weeks later), it attributed the request for a formal review to SEEPP (rather than to the Steering Committee).

[36] "Council is being asked to formally adopt this Code of Ethics per Leonard Tripp's…letter. Gotterbarn suggested that having a long delay for some minor changes to the Code is not advisable."

[37] www.acm.org/key.people/ar/cope98 ((3/19/2004).

[38] Email (Gotterbarn to Davis), March 17, 2004.

[39] Gotterbarn does not recall the exact date, but it was probably May 14 (Thursday) because, on that date, both Gotterbarn and Tripp seem to have opened their first files concerned with balloting—at about 11:34 AM (PDT). See, for example, Gotterbarn\Version 4 IEEE Vote\aBallot_ethics. Why did Gotterbarn not contact Tripp earlier that week, say, late morning or early afternoon, Monday, May 11? That would seem to be the most likely time for Gotterbarn to call Tripp, the first working day after Gotterbarn's meeting with the ACM Executive Council. Gotterbarn now had every reason to move quickly. The only reason for him to wait even till mid-day on Monday to call was that he, in EDT, would have been calling Seattle, in PDT. Of course, that he called on May 11 is one thing. That he actually reached Tripp right away is another. There may have been several days of "telephone tag" (as with Carver), ending on Thursday morning (May 14). That would explain why nothing happened until May 14—and then why a lot happened very quickly.

[40] Email (Gotterbarn to Davis), March 21, 2004 (attachment).

[41] Email (Gotterbarn to Davis), March 21, 2004 (attachment).

[42] Gotterbarn\Version 4 IEEE Vote\Ballot. The originals of the other two seem not to have survived. Since Gotterbarn had begun to use attachments, when he saved a revised version, he would have wiped out its predecessor (while preserving original file information).

[43] Gotterbarn\Version 4 IEEE Vote\ABALRSP1.

[44] Gotterbarn\Version 4 IEEE Vote\aInvitcmt.

[45] Gotterbarn\Version 4 IEEE Vote\AAVOTERS. This is an intermediate list. What seems to be the final list is: Gotterbarn\Version 4 IEEE Vote\AAVOTOUT. The covering memo (addressed to "Leonard") reads:

> Here is a text file of addresses of possible balloters. It includes names from several email lists related to software engineering and some related to ethics. It also includes the ethics officers from some Multinationals, e.g. Higgins from Boeing. Twenty six of the emails are for the SEEPP task force members.

This file was, it seems, opened on May 14, about 3:45 PM (and so must have been revised after the May 15th email in which Gotterbarn asked advice. (Yes, Burnstein, Weil, and I are on this list—with the right email addresses.)

[46] This was, of course, technically inaccurate. Neither SEEPP as a whole, nor its executive committee, had recommended a formal review process; it was the Steering Committee that had done that, with Gotterbarn eventually going along. What SEEPP (or, rather, its executive committee) had recommended was that the executive bodies of the two societies approve the Code without delay.

[47] I point out the use of the personal pronoun "we" here because that is its only use in a document that otherwise seems to speak in the name of the "IEEE and ACM".

[48] IIT's archive seems to have stopped receiving messages through the listserv sometime after December 18, 1997. This does not seem to have been caused by any change of address. I was receiving email from Mechler as late as July 7, 1998, at csep@charlie.acc.iit.edu (and Gotterbarn's May 14, 1998 list gives the correct addresses for everyone at IIT).

[49] Donald Gotterbarn, "Two Computer-Related Codes", *Perspectives on the Professions* 19 (Fall 1999): 4-6 The whole issue was about writing codes of ethics.

[50] Mechler\SEEPP98.

[51] Email (Gotterbarn to Davis), March 17, 2004. Gotterbarn added, "I did not consider the headers useful so I deleted them." So, we do not know even how many emails we should look for.

[52] Mechler\SEEPP98. There are at least three of these emails: two on 06/05/98; and one on 06/15/98.

[53] Compare Gotterbarn: "For all I know 100 people from Microsoft could have wanted to ballot but only two, that I know of, were on the official balloting group. (I don't know the rules or how it is done but they are very scrupulous to have a broad group.) They applied the same techniques for ballot group selection as when they are dealing with major technical standards like a communications protocol." Email (Gotterbarn to Davis), March 21, 2004 (attachment). For Gotterbarn, the IEEE Standards process is a blackbox.

[54] Mechler\SEEPP98. Gotterbarn then went back to the subject of what Mechler might have missed concerning SEEPP, repeating information he had already given Mechler several weeks

before: “I am working with the president elect of ieee-cs [Tripp] to follow their procedures for doing the ballots—following out their formal process.”

[55] The address problem was not “fixed”. On July 10, Mechler again wrote Gotterbarn about it (in an otherwise joking note): “I voted yesterday, guess which way? Attached is a WORD 6.0 of the short version [of the Code?] suitable for framing, vertically or horizontally with some changes in margins. It fits nice in a 8x11 frame in my office. If you want you can pass it around, I am still having trouble with my e-mail address.” Mechler\SEEPP98.

[56] Mechler\SEEPP98. “Gotterbarn\History of SE Code\History expanded” gives the date for the end of balloting as July 15 (perhaps relying on Tripp’s original schedule rather than the final one).

[57] Or, perhaps, the number should be 146. Two balloters (Samaras, #12 and #24, and Swearingen #114 and #135) each appear twice in the list of 148. Gotterbarn\1998\IEEE vote replies. But the covering memo also lists “Robert J. (Rob) Schaff” (sic) as having voted twice. He is in fact not listed even once among the 148 (though “Schaaf” does appear as #75 and #76). So, the number actually voting might be 145. Given this range, and because the exact percentages do not matter, I will use the IEEE numbers even though they use 148 as the base.

[58] I reached the low of 7% by going through the balloters names, looking for any that appeared on any of Gotterbarn’s lists (not only of earlier survey respondents but of the original SEEPP working group members, correspondents, and so on). I found eleven names (out of a possible 148). I found four SEEPP members who might have voted on Version 3 (three of whom certainly did): Burnstein, Little, Mechler, and Langford. I also found six others (not SEEPP members) who voted on Version 3 or at least commented on it: Berleur, Black , Harding, LaRue, Maner, and Sigut). (That means that ten people—just under 7%—certainly did vote both times.) I also recognized three other names not on any of Gotterbarn’s SEEPP lists: Joe Herkert (an engineer who has become interested in engineering ethics); Judith Perrolle (whom Gotterbarn knew, if from no where else, from working on the ACM code of ethics); and Rob Riser (Gotterbarn’s colleague at ETSU and on COPE). Because Gotterbarn might have urged them to vote, they too might have voted on Version 3 (bringing the total to 13, almost 9%). Since there were about thirty who voted on Version 3 but had no comment (or, at least, no comment that survives with a name attached), including Herkert, Perrolle, and Riser, it is possible that the overlap could be as high as 40 (10 + 30) names (about 27%).

[59] I have inserted “should have” because we have evidence that at least two voters seem not to have realized that comments would be public. One gave his confidential ACM voter number. Another, after voting for the Code, mentioned in his comments that he did not find the Code helpful in dealing with an ethical problem he faced:

> I have recently agreed to serve as treasurer for my church. I have been assured that the position involves overseeing assistants that are well qualified to perform the clerical parts of the job. In particular, the pastor’s wife will do the payroll on bootleg software.

Gotterbarn had both the ACM number and this story removed from the comments before posting them on the IEEE-CS web site. (I have achieved a similar protection of confidentiality by not saying which of the 148 was the source of the story.) Gotterbarn\1998\IEEE Vote Replies.

[60] Gotterbarn\1998\IEEE Vote Replies (#5, #11, #48, #52). We have the earlier comments from Berleur, Sigut, and Black, but know of Harding's only because he referred to them in his ballot comments. For Berleur's, see Chapter 8.7 (including notes); for Sigut's, 11.1. Black's comments on Version 3, fourteen pages long, are in Gotterbarn\Version 3\Comment\Black42. Based on them alone, I would not have counted Black as an (overall) negative on Version 3 (just a careful reader, thoughtful critic, and devoted supporter of a code of some sort). I count him as an overall negative on Version 3 only because he does not indicate, in his explanation of his negative vote on Version 4, that any change from Version 3 affected his vote. The negative vote seems to have been his way of making sure Gotterbarn at least considered one particular point he had made about Version 3 as well as many new comments concerning Version 4. As he said in his Interview, October 11, 2002:

> I sent in comments on V. 3 after seeing that blurb. I also commented on V. 4 after receiving an invitation to do so from Donald Gotterbarn. Most of my comments were on presentation, for example, why use the word "third party" in a few places when the text would be clearer if it used one or more of the terms already standard throughout the code ("public", "user", etc.)? Only a few of my comments were substantive. In a few places, I thought they had gone beyond what was appropriate for a professional code, for example, by calling upon software engineers to volunteer for civic work [in Clause 1.08]. I liked the code overall, especially, it saying that it was intended as "guidelines" to help you think through the problems you'll face.

Black does not sound like a "true" negative. Nor did Gotterbarn so consider him. The version of Black's comments on 3 that Gotterbarn preserved has this header (in bold): "THESE COMMENTS ARE WORTH A REVISIT ! DG".

[61] Unfortunately, I have only memory to testify to the *Queen Mary* simile. Mechler's Interview, June 11, 2002, contains only the following exchange (under Question 8) that makes the same point in a less colorful way:

> **MD:** Did you participate in any of the other versions?
>
> **EM:** I saw them, but I didn't have any problems with them. I sent them around to the committees. Really, the only time anyone complained was when the condensed version was put out, and that was really the only thing. At that time, and I don't know how anyone else felt, but we felt that we did our job.

[62] Interview of Jayaram, February 25, 2003 (under Question 17).

[63] The oddly named Junk was not alone in making this objection. Gotterbarn did not, however, follow Junk's straightforward suggestion. Version 5.2 now reads: "1.03 Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good." The sentence tacked on the end (in violation of the general format of the clauses) seems to be Gotterbarn's response to Junk. But it does not seem to me to resolve it (unless one is a utilitarian). There might, it seems, be times when we should not diminish quality of life, diminish privacy, or harm the environment even when, all things considered, the public would benefit (a little). We should not do that, for example, when the small public benefit is achieved at the cost of substantial harm to a few individuals or the environment.

[64] Gotterbarn eventually had 185 distinct responses. See email (Gotterbarn to Davis), March 16, 2004 (attachment), apparently, an email to Tripp, giving SEEPP's history through the end of 1998. The number of distinct responses is given at 8/31 (the end of the formal review process).

[65] SEEPP's Guide to Operations 4.3.5 (apparently following 1993 IEEE Standards Operations Manual) set the percentage necessary for passage at 75%, much closer to Version 4's 77% but still on the right side of the line. The Guide also set 75% as the minimum return rate for the balloting to be valid (while by 1998 the IEEE was using 60% for the minimum return rate). Gotterbarn\94-96 Misc\OPGuide feb96.

[66] While at work on Version 5, Gotterbarn received a cheering email from Prinzivalli:

> Has there been any plans developed or considered about how the Societies (IEEE/CS & ACM) are going to disseminate the Code? I have some ideas about this state of the project and am willing to work with others of the same ilk to develop the follow-on training; e.g. strctures of class room or workshop sessions, Code training session-leader, and shared omnibudsman tasks for the Code's Publication and Implementation Plan.

Gotterbarn\Version 5\The code and beyond.

[67] Gotterbarn\History of SE Code\History expanded: "8/12/98 v 5 to Tripp". There is a file for Version 5.0 with that date on it (Gotterbarn\History of SE Code\ SEEPPV5toTrip). The corresponding file for comments is dated "8/13/1998" (Gotterbarn\History of SE Code\REPLY). SERPLYDN (with file date 8/12/1998), though virtually identical to REPLY, seems to be an intermediate document of some sort.

[68] Gotterbarn\1998\IEEE vote responses.

[69] Gotterbarn\Version 5\Leonard IEE [sic] Vote Code of Ethics.

[70] Not many of the working documents for Version 5.1 survive. The only ones I have found are: Gotterbarn\ by\1998\Version 4 after IEEE vote\reply Aug 13; and (apparently identical) Gotterbarn\ by\1998\IEEE Formal Review\Jacques Berleur committees comments.

[71] Or, perhaps it went to everyone who had initially been invited to vote, whether they voted or not. What suggests this possibility is a “NOTE” near the end of the letter: “A person who did not respond to initial ballot may not raise objections to any part of the draft code of ethics which has not been changed from the initial ballot.” They may, then, still object to something new—even though they did not even vote Abstain the first time.

[72] While this email continues to calculate percentages for the original vote using 148 as the base, it uses 146 (not 145) to calculate the percentage of resolved and unresolved negatives—and that final 86.3% affirmative (which would be 85.1% using the 148 base).

[73] Sigut was not at all happy with this webpage. On September 11, he wrote Woods (in part):

> The quality of the "Comment Disposition Report" page is so substandard, that I am ashamed to refer anybody to it. If you want to see what it could look like, check "http://www.awu.id.ethz.ch/~sigut/Comment.html". In that document, I changed my Response (nr.11) so, that it corresponds to the original form, and the spurious errors (0A, 0D, !, ...) are deleted. As a side-effect, the file is ca. 3'320 bytes SHORTER, but that's another story.

Sigut saved the webpage and, following his interview, sent me the file (From Sigut—Software Engineering Code of Ethics Comments.doc). The file consists of a brief report and the spreadsheet. The report is clear, but the (crucial) spreadsheet is virtually unreadable (though one can make it readable by copying to a blank document and playing with it there). Sigut’s file did not include the ballot below the spreadsheet, though he had a problem with that too: “I do not see, how I can ‘complete the questionnaire on the ... Web page’, it being a plain text.” Apparently, even the IEEE has (or, at least, had) trouble integrating software and making it do what it wanted.

[74] The website displayed the following documents: Recirculation Ballot; Code of Ethics (Version 5.0); Summary of Changes; Comment Disposition Report; Ballot Group Member List; and Questionnaire. The Summary of Changes has survived (Gotterbarn\Version 5\Changes) as has the Comment Disposition Report in something less than its final form (Gotterbarn\Version 4 IEEE vote\sevote5a\SEVOTErpl). Among the changes the summary reported were a great many concerned with “grammar” (for example, changing “judgement” to “judgment” [undoing the spelling introduced while Gotterbarn worked in England], “consistent” to “consistently”, “methodology” to “method”, and “literate” to “well-written”). Among the substantive changes was the addition of “management of maintenance” to Principle 5, of “maintenance and support” to 1.05; the addition of “privacy” in 1.03; the substitution of “uncertainty” for “risk” in 3.09 and 5.05; the substitution of “Be encouraged to volunteer” for “Volunteer” in 1.08; and the addition of a “non-discrimination” clause (“8.07 Not give unfair treatment to anyone because of any irrelevant prejudices”). The difference between Version 4 and 5.0, though significant, is tiny

compared to that between Version 1 and 2, 2 and 3, or 3 and 4. The IEEE Standards process was bringing the drafting process to a close.

[75] The number had actually been 33, with Schaaf having voted twice. He was now listed once. The number should, however, have been 31 because G.M. Samaras is still listed twice (unless he is distinct from George M. Samaras). I should add (for those familiar with machine politics) that there was confusion at IEEE but no hanky-panky. G.M. changed his vote; George did not vote this time. Gotterbarn\Version 4 IEEE Vote\Counts.

[76] Reading these negative votes is not as easy as it may seem. When I interviewed Sigut (September 27, 2002), he seemed to like the Code *on the whole*. When I asked what he didn't like, he produced this list ("based on version 5.2 of the Code"):

> Full version; PREAMBLE; 1st paragraph; 2nd sentence: The text "are those who" could be left out. As it is, the sentence can be interpreted as a definition of a "software engineer" instead of (as I understand it) an enumeration of the software engineers' contributions.
>
> Full Version; PREAMBLE; 3rd paragraph; 2nd sentence: It still says "The list of Principles [...] is not exhaustive". Apart from my critique, I understood during the interview that this is NOT the intended meaning.
>
> Looking at the Clauses 1.04, 6.06 and 6.13 I ask: "who will protect the Software Engineer who reports or disobeys the law"? If the backing is not guaranteed, acting according to these clauses (or the given exception) might lead to unpleasant results.
>
> Clause 1.02: I would have chosen "temper" instead of "moderate".
>
> Clause 2.01: I would suggest using "frank" or "candid" instead of "forthright".

[77] Mechler\SEEPP98.

[78] Mechler\SEEPP98. Version 5.1 differed from Version 5.0 only in "some minor editorial" corrections. Gotterbarn\1998\ACM Council Response. A Word document comparison revealed the following differences: "client or employer" in Principle 2 has become "client and employer"; two sentences in the first paragraph of the Preamble have been revised a bit; the word "quantitative" has been inserted after "realistic" in 3.09; "outcome" in 5.05 has been pluralized; "better" has been deleted from 5.09 ("taking a better position"); the credits have been reorganized; and (perhaps most important) a copyright in the name of the "SEEPP Executive Committee" has been added.

[79] Mechler\SEEPP98.

[80] Gotterbarn\version 5\October 23 Passed.

[81] Gotterbarn\1998\ACM Council Response.

[82] Email (Gotterbarn to Davis), April 2, 2004. Comparing Version 5.1 with 5.2 suggests that the grammarian ignored the split infinitives (such as that in 3.07) but suggested changing "consistent" to "consistently" in Principle 1, adding a "shall" after the "and" in Principle 8, changing "consider" to "recognize" in 6.11 and 8.09, and other minor changes of the same sort. The grammarian definitely did not perform the review for consistency and style I had recommended the year before.

[83] On November 30, Mechler emailed Gotterbarn, "Did IEEE [accept] the code? You said they would look at it in November." An hour later Gotterbarn answered, "ieee-cs approved the code 'subject to legal review'. I thought I would wait on announcing to the group until the legal eagle said it was ok. More later." Mechler\SEEPP98.

[84] Email (Gotterbarn to Davis), March 16, 2004 (attachment). Actually, the date on the memo is "1/18/98" (rather than "99"), an error common soon after the change of year. The January memo also includes Gotterbarn's first attempt at a timeline for SEEPP's five year history.