Chapter 10: Slogging Toward "Version 4.DONE"

"Build a system that even a fool can use and only a fool will want to use it."
—Murphy's Technology Law # 49

10.1 Another worrying silence

About the time Version 3.0 (9. Appendix) appeared in the October 1997 issue of *Computer*, Gotterbarn began to receive comments from (more or less) ordinary members of the IEEE-CS and ACM. The first arrived on October 11.[1] It was from Stephen Unger. Then chair of Columbia's Department of Computer Engineering, Unger had long been prominent in engineering ethics.[2] In part, his prominence was scholarly. He was the author of *Controlling Technology: Ethics and the Responsible Engineer* (1994) as well as of several much-discussed articles.[3] In part, though, his prominence was practical. Long active in IEEE, Unger had an important part in drafting three codes of ethics. The first, the IEEE code of 1979, had been replaced (in 1987) by the second he worked on. That one survived only three years, its short life a record in the history of codes.[4] What replaced it was the much shorter (ten-commandment-style) code still in force in 1997 (and today). Unger had no part in writing that one. Between these codes, he had helped to write one more, the innovative but soon-forgotten 1984 code of the American Association of Engineering Societies (AAES). Unger had bad luck with codes.

Unger found Version 3 "an impressive document, full of interesting, important points, covering a wide range of topics in considerable detail." Having noted that Gotterbarn's committee had "clearly decided to go for completeness as opposed to brevity, the opposite of the choice made by the IEEE, he expressed some satisfaction (in parentheses) that "the IEEE Ethics Committee plans to attempt to get the best of both approaches, by developing, over a period of time, an extended set of guidelines for the [1990] IEEE code."[5] Then, after apologizing for not giving the Code the "careful study that it deserves", he made ten specific suggestions that together give the impression that he must have read the code carefully. (He also referred Gotterbarn to the 1991 Swedish Ethical Rules for Computer Professionals that he had appended.) Unger's first suggestion gives a good indication of the rest: "Although your item 1.07 ['Ensure realistic estimates of cost, scheduling, personnel, and outcome on any project on which they work or propose to work and provide a risk assessment of these estimates'] is similar, the wording of item 3 of the Swedish code is very nice and perhaps might suggest some adjustment in your statement…: 'only take part in projects with the time and resources to do a good job.'" (Unger's codes tended to sound as if Hemmingway, rather than a corporate attorney, had done the writing—they were always easy to read.) Though Version 4 did not incorporate this suggestion, it did incorporate several others, including Unger's second: 2.06 ("Be fair and truthful…) should instead be "Be fair and avoid deception" because, as it stands, it seems only to prohibit falsehood. The revision would make it clear (as, apparently, "fair" does not) that a software engineer should also avoid deception by "not disclosing information, in other words, deception by omission."[6]

The next set of comments arrived one day later, October 12. It was the first to mention *Computer* as the source of the code. The respondent, a member of both the ACM

and the IEEE-CS, wrote from Cedar Rapids, Iowa, that he "generally" favored Version 3.[7] But he did have "some trouble with the wording of 3.07 ["Refuse to participate in any decision of a governmental or professional body…]"—enough trouble to vote "uncertain" on that provision (and only on that one). He then filled the rest of the one-page email with an explanation of what troubled him. It was what had troubled Notkin about the same provision—except that the respondent in Cedar Rapids thought the provision too narrow as well as too broad. Limiting 3.07 to "governmental and professional" bodies would leave "a loophole for company based activities." He therefore recommended shortening 3.07 to something like "Avoid conflicting financial interests." He then gave an example of the situations that he meant this revision of 3.07 to cover:

> While I was contracting to a computer manufacturer, that manufacturer was asked to co-develop a product for a health care provider. Our allegiance fell among several employers, customers, and users. However, it was possible to render opinions on equipment and standards to each company involved without feeling bias for the companies who were not important at the time. Our group was placed into a trusted position and all participants knew the relationship. This sort of conflict may easily result when a single employer and product relationship is not established. This is often the norm these days.

The respondent from Cedar Rapids did not explain what troubled him about 3.07 or why it was not resolved by 3.06 ("Disclose to all concerned parties those conflicts of interest that cannot be reasonably avoided or escaped and aspire to resolve them"). Perhaps he did not think there was any reason to "aspire to resolve" them in the example he described. Why then did he want to broaden 3.07 (since broadening it as he proposed would make 3.06 pointless and rule out the very arrangements he described so approvingly)? Clearly, there was something wrong with 3.06 and 3.07, either individually or in the way they interacted, but what was not clear was what was wrong—and how to fix it. The problem was not logical, but something in the way (some) users would read the clauses.

Then, for almost two weeks, there was a disturbing silence—not only no comments but also no ballots. The instructions in *Computer* had been clear. All ballots and comments were to go to Gotterbarn. What explained the silence? One possibility, not a very likely one, was that no one else had noticed the code because all reference to the code, ballot, and accompanying explanation had, unaccountably, been omitted from the table of contents. *Computer* had already undertaken to correct the omission by printing everything again in the November issue—this time being sure there was a reference in the table of contents. And, of course, the *Communications of the ACM* would also be publishing the code, ballot, and explanation in November. SEEPP was now definitely a month behind schedule and, without a reasonable number of ballots, definitely in trouble. The Steering Committee would have little interest in continuing with the code without a strong vote from the membership. Three years of work would be lost. Yet, there was little Gotterbarn could do now but push on, hoping that the comments would soon pour in.

10.2 From silence to SEERI

On October 23, the first ballot showed up in a three-page email from Omaha, Nebraska (with contact information at a corporate address). A page and a half of the email consisted of a long double column: on one side, the clause numbers (without the principles); and on the other, SF, F, or one of the other answers the survey had allowed. Most of the clauses had an F after it (Favorable). Almost two dozen had SF (Strongly Favorable). There were six U's (Uncertain).[8] There were, however, two O's (Opposed) and even one SO (Strongly Opposed): 1.12.[9] Following the ballot were comments explaining one U (2.01) and all the negative votes (except for "8.08 Consider code violations inconsistent…"). Some explanations raised important points. For example, our Omaha respondent offered three reasons for voting against 1.12 ("Delete outdated and flawed data…")—the one Strongly Opposed:

> First, with respect to the data, flawed or outdated data can still be meaningful in other contexts. Second, although it is tempered with the "where appropriate" clause, the action of the statement has a direct impact on the data and places the Software Engineer in a precarious position. Data are the persistent artifacts of application. Generally, the user of the software or other individual should be the steward of data content, not the Software Engineer. Finally, if a statement regarding data quality is to appear in the Code of Ethics, it should place the Software Engineer in an advisory role to the steward of the data, not in a role to act upon the data. A more acceptable statement would be something like: "Promote understanding and awareness of the data quality and data management issues that surround the data that affect the product."

Clause 1.12 began with Version 2. The corresponding provision in Version 1 ("Assure that raw information used in software is accurate, derives from a legitimate source, and is used only in ways properly authorized") was not subject to any of these three criticisms. Our Omaha respondent (without knowing it) also objected to other ways in which Version 3 differed from Version 1 (without knowing anything about Version 1). For example: He proposed removing from 2.01 ("Disclose to appropriate persons or authorities any actual or potential dangers to the user, a third party, or the environment…") exactly what SEEPP's executive committee had added:

> Maybe it is just the phrasing of the statement, but I would strike the final phrase ("or merely know about") from this statement. I also would strike 'actual or potential'. With these corrections, I would have responded "Favor" to this item on the survey.[10]

The Omaha respondent seemed to have no objection to the substance of the provision, merely to its tone or perhaps just to its legalistic precision.

Not all of his comments were reserved for ways in which Version 3 differed from Version 1. Three of his comments concern ways in which Version 3 resembled (or was identical to) Version 1. The most interesting, perhaps, concerned 2.07 ("Not put self-interest, the interest of an employer, the interest of a client, or the interest of the user

ahead of the public's interest").[11] This is (along with Principle 2) the Software Engineering Code's (somewhat weakened) version of what engineers call "the paramountcy clause" ("Hold paramount the public health, safety, and welfare"). In some form or other, it has become a standard feature of engineering codes both in the United States and elsewhere. The Omaha respondent nonetheless objected:

> I think this statement is problematic. I think that it is important that self-interest not subvert the interests of the employer, client or public; however, the statement is too ambitious when it promotes the public interest ahead of that of the employer. If the Software Engineer has adequately "prequalified" the employer or client and determined that the employer or client is not actively working against the public interest, then I suggest that it is the responsibility of the Software Engineer to advise the employer or client regarding any concerns relative to the interest of the public at large. In this manner, the Software Engineer can play the role of educating those with the ability to affect the public interest.

Since our Omaha respondent thinks of software engineers as mere advisors, it is hard to see why he opposed 2.07. I do not think he is suggesting that it is enough for a software engineer to point out the public interest to a client or employer while recommending a course of action contrary to that interest.[12]

While most of the suggestions of our respondent from Omaha seem intended to weaken or at least soften the tone of Version 3, one does not.[13] His objection to 4.07 ("Inform when the project becomes problematic…") is that it is "too weak". He thought that, "[in] addition to informing, the Software Engineer should provide additional substantiation for his or her opinion." It is important that the software engineer "maintain a documentation regarding the informing of the client or employer".

That was the last comment for a week. Then, on November 2, there was a brief email from someone concerned with "assistive technologies". After saying that "[this] looks like a great effort and I wish you well", he quickly got down to business. He wanted to amend 2.05 ("Endeavor to produce software that respects diversity…") to include "sensory access" or "sensory abilities" (among the issues specifically mentioned "language, different abilities, physical access, …"). As 2.05 stood, he reported, he had "to read very broadly to understand that 2.05 includes the needs of the visually impaired and the hearing impaired." He concluded with thanks and the hope that "when I review this more in detail, perhaps I'll send you more comments.[14]

Almost another week passed before the next comment arrived, this one a page long from a young faculty member in Computer Science at the University of North Carolina. He was careful to say that the comments were on Version 3.0 "as published in the November 1997 issue of *Computer*". He then made ten comments, each signaled by the reference number in the code. Two were simply suggestions for amendment (with specific wording but no explanation): for both 1.05 ("ensure appropriate methodology") and 4.02 ("Ensure…document…approved for use"), "add: except with the knowledge and consent of all parties." Some suggestions indicate a problem as well as offering an amendment, for example "1.07: Again [as in 1.06], this is management's job. If the word "Ensure" were changed to "Provide", then I'm happy. Otherwise, I strongly disagree." [15] Some comments express doubts of one sort or another. For example, he said of 5.03

("Assign work only after taking into account appropriate contributions…"), "I don't understand what this means." In only one place did he seem to misread the code. Of 5.07 ("*Offer* only fair and just remuneration"), he wondered, "Why would I not accept whatever the hiring company wants to pay me, so long as the payment is not illegal and does not require me to do something unethical. What would be unfair renumeration [sic]?" Because he misread 5.07 (supposing it to apply to acceptance of remuneration rather than to offers), he said of the always troublesome 6:07 "Redundant with 5.07, and confusing for the same reason." He then took much of the sting out of these comments by ending his message:

> Good work putting this all together! This is a tremendous task, especially with all the opinions there will be from various people and groups. This has the potential to really state a direction and push SEs into more right-thinking action; perhaps even to modify behavior within software design houses to the benefit of us all. Thank you for putting in the hard work to help bring this about.[16]

Then almost another week passed without a comment on the Code. Gotterbarn had less time to worry about the silence than we might at first suppose. He was, of course, again teaching full time, but he was also setting up the "Software Engineering Ethics and Research Institute" (SEERI). SEERI was in part a way to promote the Code once it was adopted. Its website would carry the Code whether ACM's website or IEEE-CS's did or did not. He had already seen the advantage this would have. Most people seemed to have found Version 3 easier to access through Rogerson's (small) website at DeMontfort than through the much larger (and, therefore, harder to use) ACM or IEEE-CS sites. Gotterbarn also planned to use the SEERI website to list companies that had adopted the code (and thereby encourage others to do the same). The site would be a way to make available research on software engineering ethics—and on related topics such as licensing, course materials for teaching software engineering ethics, and ethics conferences. The website might even provide a place for ethics-related humor (such as Sam Redwine's quip, "Software and cathedrals are much the same. First we build them, then we pray!"). SEERI could offer seminars on software ethics, consult with individuals, organizations, and government, and even undertake major research on software engineering ethics.[17] East Tennessee State was so pleased with SEERI that it had freed Gotterbarn from the usual administrative duties of faculty and provided a graduate student to maintain the website and otherwise help with SEERI's day-to-day operation.

10.3 A positive response

While waiting for the flood of ballots to begin, Gotterbarn also wrote his task force (using the listserv) to announce that "work on the code must start again". This November 11 email reported the publication of Version 3.1 (and what had gone wrong in October), noted that the "articles" in both *Computer* and *CACM* contained "a survey for each item in the code" and that "responses are already coming in", and observed that "we should feel good about our product." But, he continued, the comments also "indicate that we are not done yet". There are "helpful comments" ranging from "things which are easy

to adjust to those which are not easy". Gotterbarn promised to organize the comments and "send them to you in a few weeks for your input."

Mechler responded (through the listserv) two days later (November 13, 1997). He had bad news: "In Computer Nov there isn't any reference in the Index. There is a Call for Participation on page 57." Even worse, "[the] web addresses on page 90 (survey) are not correct (same as on 57)." When he went to "What's New on ACM[,] Third one [presumably, Version 3] will not come up." Mechler ended by asking whether the survey is on a web page. Less than two hours later, Mechler answered his own question: "The third one is [at] the Centre for Computing and Social Responsibility [Rogerson's organization] and their correct site is http://www.ccsr.cms.dmu.ac.uk/[.] The article had it mistakenly as ccsr.cAs…" (The capital A indicates the error: it should be a small M.) Mechler concluded by reporting that the ballot "is not on any web page". Only the code is.

Gotterbarn immediately forwarded this bad news to the Task Force, preceding it with an unrelated request for help.[18] The Steering Committee had asked him for some responses from "corporate players, in addition to the individual responses." He was therefore asking any member of the task force who knew someone who could "speak for a company or a major area of the company such as systems development" to get the code and ballot to the appropriate person. Why the Steering Committee should have asked Gotterbarn to recruit corporate responses is not clear. The Steering Committee had several members who would at least seem to have had much better corporate contacts than SEEPP's executive committee (Cabrera, at Microsoft; Frailey, at Texas Instruments; Tripp, at Boeing; and so on). The same Steering Committee that only a few weeks before had sought comment directly from the TCSE and SIGSOFT now wanted Gotterbarn, at East Tennessee State, Miller, at University of Illinois-Springfield, and Rogerson, at DeMontfort University, to seek comment from corporate America. Was the Steering Committee also seeking such comments? Had they tried during the intervening month to get such comments and failed? Or were they just raising one more hurdle over which SEEPP would have to jump to bring work on the code to completion? Gotterbarn could only wonder and try to do what was asked.

Gotterbarn received the Steering Committee's request several days before he passed it on to the task force on November 13.[19] On November 18, Mechler emailed the appropriate person in his company, asking him both to respond as Gotterbarn requested and to pass on Gotterbarn's request to other chief information officers (CIOs) he knew. On November 20, Mechler's CIO wrote back with one name (and an email address).[20] Mechler then wrote (using the listserv): "One on the way CIO of Equitable Resources, Inc. Talking to another two." That was the last help Gotterbarn got from the task force in recruiting corporate response. By the end of December, Gotterbarn had several other corporate responses; some recruited from the contacts he was developing for SEERI, others just arrived unsolicited.

While Gotterbarn was making up his mind what to do with the Steering Committee request, he received a second ballot, this one with a Texas Instruments (TI) address (November 11).[21] The ballot had only four U's (and no O's or S0's): 1.12, 2.02, 2.05, and 3.08. There were only three brief comments. Though favoring 1.09 ("Ensure adequate testing, debugging, and review of software and related documents on which they work), our respondent at TI thought that "[ensuring] adequate testing and debugging

is partly the responsibility of the product provider but more so, the responsibility of those purchasing the product."[22] There is no explanation for the U vote on 1.12 or 2.02,[23] but concerning 2.05 ("Endeavor to produce software that respects diversity…"), our respondent at TI objected, "Targeting the audience of software users should be more of a concern/goal than trying to attack world-hunger with each software produced…it's a bit to[o] forward looking." What had brought on that reference to "world-hunger"? Was it the Code's explanatory sentence following the clause itself ("Issues of language, different abilities, physical access, mental access, economic advantage, and *allocation of resources* should all be considered")? Clause 2.05 was, Gotterbarn thought, one of the ways Version 3 had improved on Version 1. Another improvement was 3.08 ("Temper all technical judgment by the need to support and maintain human values"). That clause, the only other about which the respondent at TI had anything to say, elicited his most negative comment: "This principle seems too ambiguous. I cannot envision restricting technological judgment on the basis of any ethical situation."

The next set of comments to arrive was the first to mention the November issue of *CACM* as the source of the code. Its author, a principal in his own consulting service in San Antonio, began by "applaud[ing] the inclusion of language that stresses the importance of properly testing software systems prior to release." Indeed, he generally liked the code. But there was one provision he did not like, one of the innovations of Version 2 that had already come in for considerable criticism, clause 2.05 ("Produce software that respects diversity…"). The respondent from San Antonio devoted a page and a half to his five reasons for deleting the clause: "1. It's vague…. 2. It's contradictory…. 3. It's pointless…. 4. It's dangerous…. 5. It's unnecessary…." He then concluded with a paragraph that epitomized the whole:

> In Stalinist Russia, few examples so capture the fallacy and stupidity of socialism as "Socialist Science". Scientists bent their knees to the Communist Party and its tyrants, pursuing all sorts of ridiculous theories because they met with the approval of the Soviet hierarchy, turning their backs on the truth. I don't want to suggest that this subsection is on the scale of that grotesquery, but I do want to point out a parallel: All science must be ethically and morally informed, but it must not take sides in political matters. Of course, I grant anyone the right to view respecting diversity, however one defines it, as a pillar of one's personal ethics; however, I ask that this code not make it a mandatory part of the ethical values of every software engineer.

Clearly, clause 2.05 was in trouble.

The next day two more ballots arrived with comments, as well as some without; the day after that, a very long response, as well as some ballots without comments; the day after that, two more ballots with comments, as well as some without; and, before long, the tickle of response had become a small stream (about 40 ballots in all).[24] By this time (November 18), Gotterbarn (with the help of an honors student his department had assigned him for the purpose) had organized the comments into a single document, thirty-one pages single-spaced.[25] Gotterbarn sent it to the task force (in two parts) on November 20 (using the listserv), with a cover memo explaining what to do with it. After the salutation ("Hi Team"), the memo acknowledged "Ed's email about Industry input" and

reported that Gotterbarn had "contacted some multi-nationals, Lockheed Martin, TRW etc and we are getting statements from them." Gotterbarn again asked the task force to find corporate leaders (a "CEO") to respond, stressing that any such leader should "indicate that they are a CEO". He stressed this, he wrote, because "[he thinks] the steering committee's intent is to specifically identify the industry view." If the CEOs do not identify themselves as CEOs, how is the Steering Committee to know it got the industry view? He then turned to the comments (with the introduction, "Time to get back to work :-)" in capitals).

Gotterbarn had for some time been getting "responses from materials in the two magazines and on the web". Some are "useful and others are not so useful". He had included everything. The presence of a comment in the document "attached" (meaning, in those last days before attachments, below) did not "indicate support of the comment". Overall, the vote on the code's provisions had been "very positive" (approval by "90-100%"). But a few clauses (six) had not done that well (that is, had been opposed by "20-25%" of those voting). He then listed the six and commented on each:

> 1.12, - I think the issue is the SE cannot just delete data
> 2.05, - this is just a hunch—in the ballot the phrase "respects diversity" is, in the US, a politically loaded phrase[.]—This is evident by one two page critique of this clause. [The commentator from Texas had made his point.]
> 6.03 ["Support followers of this code"], Many uncertain votes (rather than oppose) put this in the 75% range—no[t] sure what the problem is.
> 6.07 ["Only accept appropriate remuneration"], There is a problem with this, we might just drop it.
> 7.02 ["Review others work only with their consent"], No idea what is wrong.
> 7.08 ["Not undermine another software engineer's job prospects for one's own personal gain"], We need to make clear that this does not rule out appropriate competition. We thought the word "undermine" would carry that weight.

What is remarkable about the provisions in trouble (apart from their consisting entirely of clauses added since Version 1 or much revised since then) is how few there are. Support for most of the code was overwhelming.

The problem, if there was a problem, seemed to be the form of the code. The chief problem was size. The most silly comment on size was, "Too long, too wordy. Can't be written on the back of an envelope." (Why would anyone want to write a code of ethics on the back of an envelope?) But there were more sensible versions of this objection, for example, one suggesting "a goal of reducing the code size to 40% of the draft size!" Some of those suggesting a shorter code seemed to do so because they thought much of the content was not about "ethics"—or, as one commentator put it: "Just because someone thinks it's a good idea does not make it an appropriate item for a code of ethics." Gotterbarn had a suggestion for dealing with this sort of comment without giving up anything: follow "the ACM model" by taking "the Eight keyword Principles and expand them by a word or three so that they could stand on their own" and attach "the clauses under each principle as 'explanatory detail', 'guidelines', 'specifications :-)'—we need a good word here." It would then be possible to print the code "in a short version, when necessary" and still "carry the sense of the whole document."[26] Having done that,

"why not call the document: Code of Software Engineering Ethics and Professional Practices[?]" After urging the task force to read the comments, especially those that "may seem ludicrous", and to suggest language to deal with them, he quoted his favorite comment on Version 3's size: "My initial reaction was 'major overkill', but I was wrong. I like it! I might even call myself a software engineer!" Gotterbarn sent the email through the listserv late in the afternoon of November 20.

10.4 How long, O Lord, how long?

The cover memo to this November 20 email had a "PS" (in capitals): "Please do not use the list address for personal correspondence!!!!" The PS seems to have been a response to an email exchange that had begun, early that morning, when Mechler had sent the listserv a one-word message "Test". An hour later, Manny Norman had responded (through the listserv): "In case you want to know—test received!" Five minutes later, he again used the listserv to announce: "By the way, I am receiving two copies of everything you send me. You may have me down twice in the database [because he had two addresses]." Norman then instructed the listserv to use his preferred address. Almost two hours later, Mechler responded (using the listserv): "First it was a test of the site; I think it was done this morning. Second, you must be on the site twice if you get more than one; I sent it to the site only. How have you been? Haven't heard from you for a while." Norman responded an hour later that he was being kept busy "juggling between performing OpenVMS system administration and programming, learning how to do same with NT, and teaching C++," adding that sometimes "I even go home! No rest for the wicked. How's yourself." Apparently, what began as a test of the listserv had unintentionally turned into a personal conversation broadcast to the entire listserv (because a "Reply" to the listserv functioned as a "Reply All").

The Mechler-Norman conversation seems to have had effects beyond Gotterbarn's postscript. Early next morning (November 21), Mark Kanko wrote the listserv—with the subject heading "Re: test": "Someone please remove me from this mail group." Even earlier that day, Tom Jewett had written Mechler and Norman (with a copy to Gotterbarn) with the same subject heading as Kanko:

> This is a large listserv, and as much as I'd like to meet everyone on it and be able to visit electronically, my inbox overfloweth even without multiple copies or test messages and personal comments that aren't meant for me. (Please note that this is NOT going to the listserv.)[27]

About 9:05 AM, Mechler responded directly to Jewett (with copies to Norman and Gotterbarn), "Sorry we inadvertently hit the reply and slowed down the large amount of traffic on the list." He then explained that he "was sending a test over and over because my e-mail kept coming back [observing that, thanks to Jewett's response, he knew that at least one person got the test]."[28]

An hour later Mechler wrote the listserv again, this time on official business, though with the salutation "Don". Mechler had not yet read "all of the attached comments" but already wondered "with such a high rating for the present form why change it now?" Changes meant "all the reviews again". Would it not make more sense

just to work on those provisions that received "negative responses"? Mechler was here exhibiting "engineering conservatism" (sometimes crudely summarized, "Don't fix what ain't broke"). But he was also identifying a deeper problem with the SEEPP process as it had developed since Melford's departure almost a year ago.

Until Melford left, SEEPP had tried (however unsuccessfully) to follow its "Guide to Operations".[29] The Guide had a well-defined process for developing "standards" (by which the Guide meant rules written, like the code, with "shall"). The working group's chair was (according to the Guide 4.1.3 and 4.1.4), to use "as expeditious a method as needed to establish the WG membership's consensus" on whether "the completed draft can subsequently serve as the foundation for an iterative process of refinement." The working group did not have to agree with every part of the document, or even be willing to have it adopted in its present form. All they had to agree on was that the document was good enough to work with. Mechler had done that with *his* "sub group" for Version 1. While Gotterbarn had implicitly agreed with Mechler's group that Version 1 could serve as a working draft, with Version 2.1 Gotterbarn achieved consensus again within his own working group (by then, also the task force).

Once there was consensus that a document could serve as the basis for drafting an acceptable standard, the next step was to enlarge the working group into a "balloting group". A balloting group had to have "balance" (that is, "[avoid] dominance by any single interest category "). The balloting group obtained balance by adding to the working group "all other appropriate individuals and organizational representatives." (Guide 4.3) The ballot put before this group was to have three choices: Approve, Do not Approve, and Abstain. Any negative vote was to "be accompanied by specific reasons in sufficient detail so that the specific wording of the changes that will cause the negative voter to change his or her vote to 'approve' can readily be determined." If the changes are made, then the vote "automatically becomes affirmative." In the absence of reasons for a negative vote, the ballot "shall, after a follow-up inquiry, be classified as 'no response.'" An Abstain counts (more or less) as a "no response".[30] At least three-quarters of the ballots must be returned (abstentions counting as returns), but more than thirty percent abstentions invalidates the balloting. The working group was then to resolve as many of the negative votes as possible (by accepting the changes proposed). Those that could not be resolved, "together with the reasons of the negative voter and the rebuttal by the WG members conducting the resolution of the ballots, shall be submitted to the balloting group, providing each member an opportunity to change his or her ballot." The Guide even recognizes that "[further] resolution efforts may be required if additional negative votes result." The Guide's process puts the pressure on the individuals objecting to find a way to agree. Where "a significant number" of individuals continue to object to some provision, the task force may recommend the standard for "trial-use", that is, for temporary adoption to see whether it can prove itself in practice. (Guide 4.3.6) Every step had a deadline. There was no endless recycling.

Gotterbarn did not answer Mechler's question ("why change now?"). Instead, an hour after Mechler posed it, Gotterbarn responded to "Ed and every body else" (using the listserv) that the question indicated that "my request to you all may not have been clear". So, Gotterbarn is not "broadcasting my reply" (a frown sign following, indicating unhappiness, whether with having to write again, or with the exchanges earlier in the day about using the listserv for messages directed to individuals, or perhaps both). Admitting

(once again) that yesterday's email (the collection of responses) was "VERY large", he tried "briefly" to clarify his request (the one in the cover memo). He wanted responses from "industry folk" identified as such. He wanted everyone to read the comments (noting that not every clause received comment). If a comment seems "on the mark" (and only then), "indicate how you agree with the comment" and also "suggest some re-phrasing". He also wanted everyone to look over the Principles to see "if we can make SLIGHT modifications to them" to pick up anything important in their subsidiary clauses (so that they can "stand alone" and still "convey the sense of the code"). If "we could do this then we could both keep the Code as a whole—Principles and Clauses[—] and print the Principles on the back of a society's membership card! :-)." After again urging members of the listserv "not to Broadcast letters that are intended for individuals", he concluded, "We are in home stretch and the results are very positive!!"

While Gotterbarn was waiting for his task force to respond, comments and ballots continued to come in. Perhaps the most unusual of these late arrivals was from the Planning and Architecture Office of St. Paul Companies (of St. Paul, Minnesota). Beginning with a very business-like "Sir", the email indicated that St. Paul "would like to adopt the ACM/IEEE code for our internal use." St. Paul had "modified it slightly to generalize the language beyond systems engineers." The writer wanted to know what sort of "source reference or citation would be appropriate". St. Paul had already been working on an internal code when it saw Version 3. Its code would gain "considerable additional strength when backed by these two prestigious societies", but it had found Version 3's wording "cumbersome for employees who are not [in] a consulting arrangement, and for IT staff that are not doing systems engineering". He would send Gotterbarn the "'modified' draft" with the revisions clearly marked.[31]

Thus began Gotterbarn's first experiment with introducing the code into a business environment. He responded "yes" (or "of course") to St. Paul's request for permission. When he contacted St. Paul's months later, he learned that the company "had only incorporated sections of the code into [its final document]." This was a success compared to some companies he dealt with later. There seemed to be two major blocks to a company just adopting the code as its own. One is (what Gotterbarn came to call) "the dinosaurs problem". A company officer who originally wanted to have the company adopt the code for its software engineers would eventually report back that the company could not "officially adopt the code" because "getting it through their bureaucracy" would be "too much of a hassle". (The "bureaucracy" made the company a "dinosaur" unable to adapt to a new environment.) The other problem was corporate counsel. Often the company's lawyer would block adoption because the company should not be "supporting one professional society rather than another." The corporation should be neutral between professional societies; and adopting one code rather than another (even if the codes covered different activities) might constitute support. One corporate lawyer raised an even odder objection. The company could not *publicly* adopt the code. The company in question manufactured computers and had a policy against revealing the source of parts used in constructing its computers. The lawyer thought that the same policy should apply to the source of the company ethics policy. The company could adopt the code, but they could not publicly admit to having adopted it.[32]

For ten days, no one on the task force responded to Gotterbarn's November 21 email—and no one else responded to the November 20 email either. Then, on December

2, Gotterbarn received two responses, one from Fairweather in England, the other from Weckert who, "still in Spain" (rather than at home in Australia), was (he apologized) in no position to get responses from industry. Both emails were long (six to seven pages). Weckert's response was all detail. He seemed to have gone through Gotterbarn's collection of comments, cutting out those he wanted to respond to, pasting them in his email, and then inserting his response in caps. A typical response would be, "Perhaps, but I'm not sure where it should be" (in response to a suggestion that the terms 'public', 'employer', 'client', and 'user' be clarified).

Fairweather's introductory sentence nicely sets the stage for the comments that follow: "My overall feeling is that if we are getting the high levels of approval that we are, we should 1) be wary of making changes that some who currently approve of the code might disapprove of; 2) be wary of major new developments such as stating the principles in a 'stand alone' manner." There followed pages of detailed response to particular changes Gotterbarn (or, more often, one of the respondents) had suggested. Of special concern to Fairweather was the short version of the code Gotterbarn had suggested. While wary of it, he thought it "would be OK **provided** we make it clear that they [the Principles standing alone] are only extracted from the full code, and should never be used on their own, or quoted without a reminder that they are extracted from the full code." Fairweather even took the trouble to suggest the form in which he thought the extracted Principles should be printed. Of special importance to him was that the in-particular clause not be omitted. The presence of an in-particular clause for each Principle would make clear that the Principle could not really stand by itself.

10.5 Three days in Baltimore

Gotterbarn was now making final preparations for a meeting with the other two members of his executive committee to revise Version 3 (or, as he now admitted, "build Version 4"). The meeting would be in Baltimore. An important step in those preparations was incorporating into the comment document all comments that had arrived since November 20. That work was completed by December 1 (the day before the Fairweather and Weckert emails arrived).[33] The final summary, though a few pages longer, contained nothing strikingly new. The comments had become predictable.

By December 3, Gotterbarn had two other documents ready. One was the final tally of ballots. Using Excel, Gotterbarn was able not only to tally the votes, but to calculate percentage in favor and opposed to each clause. Overall, the mean approval had been just over 94% and the mean disapproval just over 3% (with the difference in uncertain votes).[34] Four clauses (1.10, 1.11, 3.02, and 3.03) received unanimous support; and one more (3.06) passed without any votes opposed. Software engineers seemed to have no doubt that they should promote the privacy of individuals (1.10), use data legitimately (1.11), only sign documents within their responsibility (3.02), and reject bribery (3.03), and were only slightly less certain that they should disclose conflicts of interest (3.06).

Having completed the tally of ballots, Gotterbarn could prepare what he jokingly called, "Version.3b mod-notkin, et al").[35] This was the Version 3 that the executive committee would work from. The first two lines, inserted above the title (in large print, bold, italics, *and* underlined), explained the format: "This has some of the modifications

[to consider] and clauses in bold have less than a 90 [%] favor rating." Twenty clauses were in bold (a quarter of the Code's eighty clauses).[36] Below this heading (in bold) was the new title of the code ("Software Engineering Code of Ethics and Professional Practices") and, immediately below that, the same title again. This was the first of the "modifications" already incorporated. Next in bold was a question, "Does this footnote get attached to the CODE?" The footnote, listing the codes "referred to", had, until then, been part of the article introducing the code. Why Gotterbarn thought it should be part of the code is unclear. No one who commented on Version 3 had suggested it should be and, of course, footnotes of that sort are rare in codes of ethics. (Indeed, I know of no code that has a footnote listing source codes.) Here perhaps the scholar had gotten the better of the draftsman. (The footnote did not survive Version 3b.)

All the modifications seem to be responses to Notkin.[37] The first (in caps) is just under the title "Preamble": "Told Notkin that this was going in the Preamble—where." "This" consists of two sentences:

> Software engineering is unique in the amount of time spent in maintaining and modifying existing software. The flexibility and maliability [sic] of software systems places [sic] special burdens on the software engineer to treat software modifications with the same professionalism as new development.

The first paragraph has two places where "and maintenance" (in italics) has been inserted after "development", with other changes that Gotterbarn had promised Notkin scattered here and there. The largest of these was in 6.13: "Share useful software-related knowledge, inventions, or discoveries with the profession, for example, by presenting papers at professional meetings, by publishing articles in the technical press, and by serving on the profession's standard-setting bodies, *provided this does not violate confidentiality, or cause loss of trade secrets, or cause competitive disadvantage (see 4.05).*" (This clause—without the addition indicated by italics—had received an 89% favorable vote, just under the 90% cut off.) There is only one surprise. Just above clause 3.07 ("Refuse to participate…") Gotterbarn wrote (mostly in caps), "Early returns had some objections to this/now sits at 95% positive! Told Notkin it was gone. Ooops."

There is a lesson here. Notkin had (rightly) said that he spoke for himself. He had made a prediction about what his technical council would do, but seems to have badly misjudged it. In fact, the council seems not to have cared enough one way or another to respond. Notkin had never had to test his own ideas in open debate. He had supposed that he spoke "common sense" (and Gotterbarn had thought so too). But the December tally at least suggested that common sense may not have been what Notkin supposed. Leaders generally suppose they know what "the profession thinks". This was, it seems, what Shaw and Frailey thought too. They might have been right—but, in fact, they were, it seems, largely wrong. There are nonetheless at least two reasons to take the views of leaders into account. One is that they may have a better understanding of the issues than their followers; another is that they may have the power to block a code even if they are wrong in their views, dogmatic, and dense. The first reason is always worth considering; the second, something to get around if possible, for example, by taking a vote that reveals how wrong the leader is in a way not easily dismissed. Gotterbarn would have done well

to promise Notkin less, especially since he did not (it seems) send his official response to Notkin until most of the ballots were in and tallied.

The next day (December 4), Gotterbarn, Miller, and Rogerson met briefly in the lobby of the Brookshire Hotel in Baltimore. It was late afternoon. All of them were tired after traveling most of the day. Each had a laptop computer with him. Gotterbarn distributed copies of the documents he had prepared (the tally of votes, the final collection of comments, Version 3b, and the emails from Weckert and Fairweather). Some of these were on a diskette; others on paper. After some discussion of large issues, division of labor, and timetable for the next three days, the three agreed to meet again in Gotterbarn's room early the next morning, adjourned to dinner, and retired early.

The meeting's location was not arbitrary. Gotterbarn had chosen Baltimore because traveling there was relatively easy for all three, because its hotels were relatively inexpensive, and because each would be far from the distractions of home. Gotterbarn had chosen the Brookshire because it had all the conveniences of the city close by, was not too expensive, and had two-room suites. The outer room of the standard Berkshire suite had a work table (or large desk)—as well as the usual couch and comfortable chairs. Only Gotterbarn's choice of meeting room was arbitrary. Miller and Rogerson each had a similar suite down the hall from Gotterbarn's.[38]

The three (as agreed) met in Gotterbarn's suite after breakfast. They were to go through Version 3b once, making such changes as the comments or tally suggested; then to go through the revised document, making sure that it said what it should and, when it did not, revising accordingly; and so on until they were satisfied. Nothing was immune from improvement. Once work on the code was complete, they would draft two letters: one to the task force explaining what they had done; the other, to the Steering Committee asking it to recommend Version 4 to the two societies for approval. Gotterbarn expected the work to take two full days. In fact, it took much of two evenings as well—and much of the third day's morning. Rogerson, who volunteered to be the recording secretary, sat at the table. Sometimes Gotterbarn and Miller sat at the table too, looking at a paper copy of this or that document or amending their own paper copy of Version 3 when they had agreed to a change. Among the pile of papers on the desk was one containing comments (and ballots) that had been faxed. These (unlike the emailed comments) were generally short (and handwritten).[39] Sometimes, though, Gotterbarn and Miller stood behind Rogerson, looking at what he was typing, or took turns pacing the oblong room, sitting on the couch or in one of the chairs. Though the room had a big window at one of its short ends, little day light came in. (Gotterbarn had pulled the drapes to make reading the computer screen easier.) Discussion was almost continuous. As Miller recalls,

> Several times we got into deep discussions about the wording of a clause. I think all three of us felt that what we were doing might have significance beyond what usually occurs in academic scholarship. We wanted to agree to something (to get it "out the door"), but we wanted to get it right. Some of the suggestions we had gotten about the code made it clear that every word counted --- there were shades of meaning in what might otherwise be considered trivial word changes.[40]

On the first day, Gotterbarn put off lunch until the middle of the afternoon saying over and over, "We're making progress. Let's just finish this first run through, then go to

lunch." Gotterbarn treated toilet breaks in the same way—until Miller revolted, declaring he was going "now" no matter what. Only when keeping track of changes to their paper copies of the code became too hard did Gotterbarn agree to lunch, and then only on condition that they stop at a copy shop to have the code's electronic version put on paper. Having eaten a quick lunch, they found a Kinko's. Kinko's copiers had only recently been put on a network that allowed printing from a diskette. The first attempt to print failed. The staff, well-groomed clerks in their early twenties, were not sure what was wrong. SEEPP's executive committee, with no time to waste, volunteered to fix the problem. The staff hesitated, perhaps worried what the three, all obviously survivals from the pre-electronic age, might do to their sophisticated system. Reason, firm looks, claims of expertise, and a few well-placed technical observations won out, however, and the three software engineers were soon checking wires and adjusting settings until they identified the problem, a cable that had come unplugged. The network fixed, they quickly printed three copies of the nascent Version 4.0 and returned to their work at the Brookshire. Miller thought it "was somehow fitting that we old geeks figured out what was wrong at Kinko's. We were going to get that code finished no matter what!"[41]

Dinner, though quite late, was pleasant. The three had a first draft of Version 4. The end was in sight. Tomorrow they would work on the details. Next day, they worked much as they had the day before—but when they went to dinner that evening, they had everything they had hoped for in the code. The next morning they completed work on the two letters. By noon of December 7, the meeting was over. They had checked out and were on the way to the airport. Reflecting on what they had done, Gotterbarn observed, "The most effective way to do it [this sort of major revision] was NOT by email from separate distracting environments. Face to face—mind to mind—working in the same direction gets the work done much better and more efficiently."[42] The three days in Baltimore had had much the same feel as those months at DeMontfort. Gotterbarn was (as he put it) "in his glory" during those three days, but for several years afterward Miller (with a smile) referred to Gotterbarn as "the taskmaster".[43] Such is the price of glory.

Gotterbarn flew out of Baltimore early in the afternoon of December 7, getting home quite late. The next morning (December 8) was the beginning of exam week. He had tests to give and projects to grade. Almost the first thing he did that morning was to send the task force Version 4.0, with the cover memo (signed "don") that the executive committee had prepared the day before. After declaring that "the results of several emails and a three day meeting by the executive committee" were "below", Gotterbarn noted, "We need to send the FINAL COPY OF THE CODE BY MONDAY DECEMBER 15." He was asking the task force to "give this one more review". The "substance of the code" was unchanged, but "we did clear up several things and restructure it." He then summarized what the executive committee had done. First, they had "modified the Principles so that they can stand alone as a reminder of the entire code". The short version now stood before the main code. Second, they had shortened the Preamble, but "also included issues of maintenance", and had tried to make it clearer "when resolving ethical tension that the public interest is paramount". Third, they had re-ordered the principles, putting "the PUBLIC first where it belongs". Fourth, they had reordered the clauses "in general putting the more aspirational elements first and the more specific items later". Last, they "clarified the function of the code by renaming it". Gotterbarn concluded by again pleading for "your comments".

Though this summary of changes makes it clear that Version 4.0 differed substantially from Version 3.0, it in fact substantially understates the differences in at least five ways. First, the executive committee had shortened the Preamble, in large part, by deleting Version 3's second and third paragraphs. All references to "three levels of ethical obligation" were gone. Comments had shown that the distinction confused rather than clarified. The idea of a code as consisting of "aspirations", "expectations", and "demands" had (at least officially) been abandoned.[44] Whatever its theoretical value, the distinction had no practical value within a code. As Gotterbarn explained years later:

> In version 3 I had tried to meet some of the objections we had heard early from the steering committee- what makes this a SE Code of ethics when it says some things said by other profession's codes? So the preamble contained comments on three levels of ethics—common human virtues, professional ethics having due care, and specific professional ethics related to a particular domain. It was this latter part that made it an SE code. Anyway, this distinction caused confusion because people reading the preamble expected to find each clause clearly labeled as to which of these three levels it was on. We knew that any attempt to deal with this type of [objection by] dividing the Code would have landed it in a mire of debate about which clause belonged where and would have distracted from the intent of the Code. One of the good things we did at that meeting was to scrap my three-level stuff.

What the executive committee meant by re-ordering the clauses so that the "more aspirational" ones come earlier is not clear. In what way, for example, is 1.01 ("Accept full responsibility for their own acts") more aspirational than 1.08 ("Volunteer professional skills to good causes…")? Is this another example of the concept of "aspiration" doing no work?

Second, the executive committee had not limited changes in the Code to clauses with a relatively high percentage of opponents. They had in fact made many small changes in clauses with very high approval ratings. For example, they had replaced "employee" with "software engineer" in both old 5.01 and 5.02 (new 5.02 and 5.03). Both clauses had been approved by more than 95% of those voting.

Third, the executive committee had also made larger revisions in clauses with equally great (or greater) support. The most dramatic of these revisions is the new 4.04 ("Do not engage in deceptive financial practices such as bribery, secret payments and double billing"). This clause replaced *three* clauses in Version 3: 3.03 ("Reject bribery"); 3.04 ("Accept no payback, kickback, or other payment from a third party to a contract, except with the knowledge and consent of all parties to the contract"); and 3.05 ("Accept payment from only one party for any particular project, or for services specific to that project, except when the circumstances have been fully disclosed to parties concerned and they have given their informed consent). Clause 3.03 had a 100% approval; 3.04, 98%. Only 3.05 (at 89%) had failed to achieve the 90% that (in Version 3b) declared would mark a clause for revision.

Fourth, while this one revision may seem to explain the drop in number of clauses from 80 to 78, it in fact does not. Several other clauses had entirely disappeared (such as old 2.03 "Affix their signature only…" or old 7.08 "Not undermine another software

engineer's job prospects...).[45] Others (unrelated) had replaced them, for example, new 3.15 ("Treat software maintenance with the same professionalism as new development"), inspired by Notkin, or new 5.12 ("Not punish anyone for expressing ethical concerns about a project"), which, though reasonable, does not seem to have come out of the comments at all.

Fifth, Version 4 was about four-fifths as long as Version 3.[46] The change is actually greater than this fraction suggests. After all, the one-fifth reduction was achieved while adding the (largely redundant) Short Version (189 words). Without the Short Version, Version 4 would have been just under three-fourths the length of its immediate predecessor (though still almost twice the length of Version1). A reduction in length of one quarter, especially when combined with the addition of new clauses, is independent testimony to the scale of revision accomplished in the first week of December. Version 4 is almost a new code.

The executive committee had, it seemed, not felt at all bound by the poll of ACM and IEEE-CS members, by the drafting principle "Leave well enough alone", or by Mechler's concern that making any but minor changes in Version 3 would bring on another poll, putting off adoption for many more months (and thereby chancing a wreck on shoals already perilously close). Whatever improvements the executive committee had made (and there were many), they came (as the committee should have learned from the large dissent on "diversity") with the risk that what they thought uncontroversial improvements might prove quite controversial. They were, in effect, gambling the overwhelming support the Code had achieved.

Re-ordering the Principles and the Clauses under each Principle also had the effect of making tracking changes, and comparing changes and votes, much harder for anyone who tried (including the executive committee). That effect, though doubtless unintended, is worth noting, especially since there was not much reason to reorder the Principles (only one commentator suggested it) and no reason to reorder the Clauses (since no commentator had suggested that). Even someone who liked the re-ordering might complain about that effect. For example, Fairweather wrote that the "[re-ordering} was good, but in the process it becomes more difficult to spot which clauses have been deleted, and some of the ones deleted, I feel, ARE important." [47]

10.6 First comments on Version 4

Gotterbarn had asked for comments when he sent Version 4 to the task force on December 8. The first comments to arrive were mine (on December 9), a three-page fax (not counting the cover).[48] My "overall comment" set the tone for the particulars that followed: "very soon, it [the code] should be gone over by a good editor." To soften the implied criticism, I added in explanation: "Committees have a tendency to garble text, to introduce unnecessarily clumsy modes of expression, and otherwise to make the code more difficult to use than need be." The comments that followed this explanation were (I said) intended as illustrations of what I had in mind; my silence on substantive issues indicated only "my virtually complete satisfaction with the Code's substance". I said nothing about the reordering of the Principles or clauses because, though I disapproved, I regarded order as a matter of taste. There are many ways to order a code. If the likely

users could not see the advantages of the order I chose, they were welcome to anything they thought better.

My objections were of another sort. The revisions had introduced what in software are called "bugs", at least three kinds of them. First, I had found a great number of simple errors (typos)—indicating nothing more than haste. For example, consider my suggestions (given in brackets) for the last three lines of the Preamble's fourth paragraph: "In these judgments concerns [delete s] for the health, safety, and welfare of the public is the primary concern…. That is [insert ,] the 'Public Interest' is central to this Code."

Second, I had noticed sloppy drafting. The first instance (occurring both in the short version's introduction and in the corresponding sentences in the long version's Preamble), was a "transition from 'must' [in the first sentence] to 'shall' [later in the paragraph]". Whatever the point of the transition, it is (I said) "too subtle for me". I therefore recommended "using 'shall' in both (since 'must' sounds too much like banging on the table)." Changing terms without a clear change in meaning invites confusion. Does the change mean something? What? (The next edition of Version 4, and all Versions 5, followed my suggestion in the Short Version but not in the Preamble.)

I also found several places where, though there was no risk of confusion, there seemed to be unnecessary redundancy. For example, I suggested revising the last sentence of the Preamble's fourth paragraph ("That is[,] the 'Public Interest' is central to this Code"): "delete this entire last sentence as entirely redundant, enough [in the sentence before] to say the public health, safety, and welfare are primary." This suggestion was not taken.

Third, Version 4 seemed to have introduced a number of conceptual problems. Consider, for example, the new 4.04 ("Do not engage in deceptive financial practices such as bribery, secret payments and double billing"): "Is bribery a 'deceptive financial practice'? [I asked] What if the bribery is open?" It might be better to say, "not engage in bribery, double billing, or other improper financial practices." I had, I added, "[omitted] 'secret payments' since some—for example, those involving national security—might not be improper."[49]

Another one of these conceptual problems was in new Principle 1. It now read, "Software engineers shall always act in the public interest…" I suggested that it was enough if they always acted in ways "consistent with" the public interest. "To require them always to act 'in the public interest' is to forbid them to do good for themselves, their clients or employers, and even customers when doing that is neutral with respect to the public interest." Surely, that was not what was intended. Indeed, that suggestion would be inconsistent with Principle 2 ("Software engineers shall act in the best interests of the client or employer within the confines of the public interest.").[50]

Gotterbarn seems to have received only one other response from the task force before Version 4 was due at the Steering Committee. That one, by direct email, was from Ben Fairweather—like me, a philosopher, not a software engineer. What he wrote was, as he put it, merely "IMHO" (in my humble opinion); it was almost entirely substantive. Apparently, Fairweather had not only read through the document carefully, he had, despite the difficulty re-ordering introduced, systematically compared each provision of Version 4.0 with the corresponding item in Version 3.0. He had much to say about Version 4.0, some negative but much of it positive. Quickly, without comment, Gotterbarn forwarded Fairweather's email to the listserv.[51] Later that day, Rogerson

(Fairweather's boss) replied directly to Gotterbarn, with a copy to Miller. A few hours later, Miller also responded.[52] Miller seemed a bit grumpy. His "initial reaction to all these comments" was that "we cannot do this forever, trying to please everyone over and over." Having made that point, he relented: "if we can QUICKLY come to some language that is a clear improvement on our already improved language, so be it." Miller did not have much to add, but what there was was inserted below the relevant part of the Fairweather-Rogerson exchange he had pasted into his email. The Fairweather-Rogerson-Miller exchange offers more insight into how the executive committee now approached the comments they were receiving.

Fairweather's positive comments showed the care he had given Version 4. Many were of the form: "Improved" (after quoting the new Principle 1 in full); or (after quoting 2.01) "An improvement on the old 4.01"; or most often (after quoting the new 2.04) "A mild improvement on the old 4.02". But some of the positive comments were more complex (and mix in criticism); for example, Fairweather admitted that the new Principle 3 ("Software engineers shall ensure that their products and related modifications are of the highest professional standards possible….") has been improved by the "removal of detail" but adds, that "loss of mention of products being 'acceptable' to the employer, the client, the user, and the public creates a deficiency with the code: it should have a clause [requiring acceptability]." [53] Rogerson's response was, "If they are of the highest professional standards, then by inference they will be acceptable as this is part of the definition of professionalism." Then, perhaps realizing the risks of relying on inferences from a definition of "professionalism" not given in the code (and perhaps controversial), Rogerson relented, "An extra clause (or addition to an existing clause) is worth doing." Miller disagreed "violently":

> If you want yet another clause, fine. I might point out, however, that you can work to the highest standards and still have some idiot not "accept" your work. Part of being a professional is knowing when to adhere to standards when idiots demand something else.

A good point, but one that admits the possibility that "the highest [professional] standards" might be too high for public, customer, or user (unless "possible" means something like "practical" rather than "technically possible"). The more detailed language of Version 1, which survived more or less through Version 3, avoided this problem: "Software engineers shall, insofar as possible, assure that the software on which they work is useful to the public, the employer, the client, and the user, completed on time and at reasonable cost, and free of significant error." Anyone who did not find acceptable a product that was useful in this way, completed at reasonable cost, and free of significant error would, indeed, be an "idiot". The new Principle 3 was not changed.

The most negative of Fairweather's comments concerned the newest part of the code: "I cannot accept this short version as it stands. If it is not modified, I will have to ask for my name to be removed [from the list of SEEPP members at the end of the code]." Fairweather then explained why he was unhappy (in effect, summarizing what he had written on December 2): "the lack of warning not to treat this short version as a complete code of ethics is a fatal flaw." (10.4) The flaw was, however, easy to remedy, just "add a line that it is not the complete code, and is only an 'aide memoire' for the full

version, which itself is not intended to be used so that individual parts in isolation could justify errors of omission or commission." Rogerson's comment was that "Ben's concern is valid". He suggested adding a sentence that "points people to the full code and explains the importance of the clauses."[54] Miller agreed and even suggested a paragraph that (without amendment) became the first paragraph of the short version of the next edition of Version 4.[55]

Fairweather's negative comments did not always receive such favorable treatment. Consider, for example, the discussion of 8.08 ("Encourage colleagues to adhere to this Code"). Fairweather objected that it "doesn't appear to me to be suitable for the 'self' principle" (and suggested going back to old 8.08 with a small revision). Recalling that I too had suggested that 8.08 belonged elsewhere (under Principle 6 Profession), Rogerson recommended instead that 8.08 be revised to read, "Consider violations of this code inconsistent with being a professional software engineer." Miller's response was, "I don't care." The next version of 4 adopted Rogerson's suggestion.

Rogerson was not always this accommodating, however. For example, Fairweather objected to the rewriting of Principle 5. The principle, while improved, had (Fairweather noted) lost "1) acting fairly (in general) and 2) 'enable and encourage those who they lead to meet their own and collective obligations, including those under this code'." Judging that loss to be "severely detrimental", he recommended adding "a new 5.01 [for acting fairly] and 5.02 [for enabling]." "Not convinced—leave as it is", was all Rogerson had to say. Miller agreed: "Yes, leave as is." It remained as it was.

Occasionally, Miller was quite dismissive. Fairweather thought it "detrimental" to delete from old 2.01 (now new 1.04) the words "or merely know" after "reasonably believe". Fairweather would accept deleting "merely". Rogerson replied that he thought "believe" includes "know" in this context. (1.04 read: "Disclose to appropriate persons or authorities any actual or potential danger to user, the public, or the environment that they reasonably believe to be associated with software or documents.") Miller responded directly to Fairweather (though Fairweather was not among the email's addressees and probably never saw the response): "Put a cork in it, Ben. What we have here is good." Clause 1.04 was not amended.

Over the weekend, Gotterbarn revised Version 4 (while also grading exams). Monday afternoon, December 15, he emailed the listserv a one-page memo entitled: "Version 4.DONE". The memo maintained the tone set by that title. Gotterbarn thanked "you all" and described the writing as "an exhilarating experience"—adding that "[when] projects come to a successful completion[,] there is normally a celebration and everyone shakes hands and is proud of the work accomplished." He expressed the desire to meet all those he had so far only known through email. "The problem with E-mail is that many of you only have a virtual persona for me." Then, getting down to business, he reported, "We have received several comments from the task force on Version 4.0." Some were positive, some were not so positive; some substantive, some concerned with "issues of grammatical preference". "We" had tried to take them into account in the "final version". Some of the lengthy changes might be better in "worked examples of the Code". (Apparently, Gotterbarn was here referring to one of Fairweather's suggestions.) He would, he said, explain those "worked examples" in a later email. The code was not perfect. But, as in any project, "at some point we must call [this one] complete even though we can still think of other improvements." He had decided now was the time.

Hoping the task force approved of what had been done, he noted that his memo was short "because I have placed the final version on a web page: http://www-cs.etsu.edu/seeri/secode.htm."[56] (SEERI was now in operation.) Task force members should read the code "one more time" and be sure to "carefully check the credits at the end of the Code" because he wanted "the form of your name and its spelling to be correct and to your liking." This version was the one "being sent" to the Steering Committee for adoption by both the IEEE-CS and ACM.[57]

10.7 Party time

The next morning (December 16), just before 9:00 AM, Manny Norman wrote the listserv (replying to "VERSION 4.DONE"): "Thanks, and the season's greetings to all of you. Perhaps, we could figure out some sort of get-together at some mutually acceptable geographical location sometime in the New Year." Apparently, Rogerson liked the idea. He quickly responded: "How about the Christmas Islands?"[58] Duncan Langford soon joined the horseplay: "Brilliant idea. Now, have we enough left in Don's budget to fund fares & accommodation…:-) Season's greetings to all!" About 11:00 AM Gotterbarn joined the discussion: "Budget ???? did I miss something?!?!" (Almost everyone who had anything to do with SEEPP knew by now there was no budget.) Having made a joke, Gotterbarn almost became serious: "There is a splendid conference ETHICOMP '98 in Rotterdam in March. The ACM is having an Ethics Conference in Washington in May. We can have two parties."[59] About the same time next morning, the horseplay began again. Mechler wrote, "I never thought e-mail could catch a sigh of relief but the last few days emails was [sic] a perfect example." Then, perhaps thinking this comment was not light-hearted enough, he continued, "Isn't it just like a leader to ask what budget when the project is over and everyone wants to celebrate." Mechler wondered how there could be an ethics meeting "in Washington????" and wished everyone "Happy Holidays". A half hour later, Norman ended what he had started the day before—by a lisping pun: "Maybe we should all meet across the Detroit River from me—in Ethics (Essex County, Ontario!)." Then, perhaps thinking he had gone too far with the pun, Norman apologized: "I know we're not supposed to be using this listserv for 'flippancy', but I think we have reason to celebrate this time."[60] Everyone must have agreed—or, at least, no one wrote the listserv to condemn the exchange. It was, after all, doubly a season to be jolly.

Before the horseplay was over, Gotterbarn sent the Steering Committee—or, rather, its chair and vice chair, the official ("final") Version 4 with a cover memo (just over two pages long).[61] Signed "The Executive Committee of the SEEP Task Force", this memo is itself a significant document. Not only did it summarize the task force's achievements in a way likely to impress the Steering Committee as a reason to recommend the code to the two societies for approval, it also announced plans for disseminating the code that seem not to depend on whether the two societies approved. The memo made no mention of the Steering Committee's plan to disband at the end of the year (a plan about which the Steering Committee had said nothing since September and which the executive committee doubtless hoped had been forgotten).

The cover memo's first paragraph offered five (unnumbered) reasons for recommending the code for approval:

- Support by both societies will be a public assertion by the profession of a unified commitment to ethical responsibility,
- The Code represents an international standard of practice,
- The Code has been reviewed and adopted by industry,
- The Code presents a broad consensus of the profession, and
- There is wide demonstrated positive support, including from Academia, for the Code.

The next few paragraphs provide detail in support of these claims. For example, to explain why the support of the two societies is important, the memo distinguished the software engineering code from those of the two societies. The software engineering code "addresses the ethical issues and standards of conduct which specifically arise within the practice of software engineering." It is not ("[in] this sense") a substitute for any existing code.

While stressing the international participation in the task force ("Europe, Africa, Australia, and North America"), the memo also pointed out other ways in which SEEPP had avoided too narrow a perspective. The task force had sought "responses from members of the major computing societies". It had "received [comments and support] from the International Federation of Information Processing (IFIP)."[62] It had even included "two lawyers [Barber and Phillips] specializing in computer law, and one representative from the military [Kanko]." (Perhaps wisely, the executive committee did not mention the participation of three academic philosophers.)

The executive committee also reported an impressive response from industry (especially impressive, given the few months available to obtain it). Several ethics officers or ethics directors of "multinational organizations" had reviewed the code "positively… after sharing it with software engineers in their organizations." Among these organizations were "Lockheed Martin, Northup Grum[m]an, and Digital Equipment."[63] The code had also received support from smaller companies: In fact, the code had already "been adapted by St. Paul Companies as their information systems code of ethics with only minor modifications, such as replacing the expression 'software engineer' with 'IS professional'."

The executive committee then described the "process of continuous review and revision" by which the final Version 4 had reached the Steering Committee. The description seems somewhat garbled. On the one hand, the executive committee wrote, "The final draft of the Code" (that is, Version 3) was published in *Computer* and *CACM* along with a survey. Most items "had greater than 95 percent support with many at 100 percent [with the highest level of opposition being 12%]". Then, having described the overwhelming support for this "final draft", the executive committee continued, "In the light of feedback, the Principles and associated Clauses have been reviewed and modified where necessary." That is, the "final draft" was not the final draft. The executive committee concluded this description of process with the hopeful (but unsupported) statement: "We are confident that if the survey were to be taken again, all Clauses would receive an approval rating of over 95%."

After a brief paragraph demonstrating academic support by "several requests to have the Code included in computer ethics and software engineering textbooks" and other requests to develop cases based on the code, the executive committee reached its

triumphant last paragraph.[64] The code had "already proven to be a dynamic and useful document", a "model" within at least one company; both the ACM and IEEE-CS had "already recognized that education about their codes of ethics is an important issue"; and several members of SEEPP are members of committees charged with educating members "about their codes of ethics". To support "education about the Code", SEEPP planned "to maintain [the code]" on two websites, one at Gotterbarn's SEERI and the other at Rogerson's CCSR. SEEPP would "continue to gather input from the community about the Code and develop and share examples of using [it]." Companies had "already donated case studies related to the Clauses of the Code." It would therefore be good for "the profession if both societies endorsed this effort and approved the Code."

SEEPP's executive committee neglected to mention that it had accomplished all this while delivering the code six days *ahead* of the schedule Gotterbarn had proposed ten months before. The executive committee was also unable to report three more pieces of good news Gotterbarn would receive just before Christmas (December 22). One was an email from a sales manager at Hitachi America's New York City office, a late ballot. The sender had voted in support of every provision, but for several provisions seemed not to have been satisfied with "SF" (strongly favorable) as a way to express accurately how strongly he favored the provision. So, every now and then there was an "x" far to the left of "SF". His only comment was that he would also fax his response.[65]

December 22 seemed to be a day for catching up on correspondence. Earlier in the morning, Gotterbarn had received an email from Bangalore. The author, who worked for Siemens Information Systems (India), wanted to know where he could "get a soft copy of the draft SE code of ethics V2.1 that you published in SEN 22 (July 1997)". Even "plain text" would do. He wanted to make it available online in his software development center for wide circulation."[66] Why had it taken so long for Version 2.1 to reach India—or so long for India to respond? Whatever the answer to that question, Gotterbarn could now add Asia to the continents participating in drafting the code.

Then, after dark, Gotterbarn received another email from Siemens, this one from Germany. Two emails from Siemens in one day? Could that be mere coincidence? The email did not answer that question. It was (like the email from Hitachi) a ballot. But the German had not bothered to print out the long columns because "I (strongly) favor the clauses put forward. I think, the code could be voted on [in] its present form [Version 3.0]." He nonetheless had several suggestions. One was that "clauses 1.04, 1.07, 1.08, 1.09, 1.12, and 1.14 should be inspected more closely".[67] They might be assembled as "an approved collection of methodologies and (possibly augmented) standards." The other comment concerned 6.10 ("Obey all laws governing their work…").[68] The German thought this clause needed clarification. "Is censorship consistent with public welfare?" He wondered whether the clause could be dropped altogether: "If it's illegal, let the government's lawyers worry about it." He even wondered whether what was meant was, "You may disobey the law, if obedience is inconsistent …"

Gotterbarn does not seem to have passed these three emails on to Miller and Rogerson. Instead, he wrote the two (on Christmas day) with a problem. But, before stating it, he described some of his recent correspondence. From the description, we can tell some have been lost. "People are," Gotterbarn began, "still sending in votes on 3.0." One of these, a "vp in Chicago", had passed out the survey to thirteen of his subordinates. "He should," Gotterbarn thought, "be upset by the results—6 of the 13 strongly oppose

'being responsible for your work'!!!!!" Gotterbarn also seems to have responded to the correspondent in India. He was (Gotterbarn reported) an "Information Systems Manager" who had read about the Code "in the July issue of SIGSOFT Newsletter". He wanted to distribute the code "to all of his employees :-)". The only version now available ["4.0"] was on the SEERI website. That, however, was not the problem he wanted help with. The problem was "the long, carefully worked out comments I get by email, some as long as 5 typed pages,—to which we [the executive committee] replied 'we will take these into account in the next version early next year'." Gotterbarn wondered whether he should instead "ignore the emails, or [answer simply] 'your comments will be taken into consideration'."[69]

Miller responded on December 27 (with a copy to Rogerson). It was (he thought) important that "the people do get a response".[70] He wanted them to feel part of the process. But any response could be no more than a response "for the three of us, since we have the authority to speak only for ourselves." Miller suggested the response make four points: First, Version 4.0 (in its final form) is now before the IEEE and ACM governing bodies and is no longer subject to change. (Correspondents should be sent a copy.) Second, "we" should "tell people (and commit ourselves to the project) that we will begin work on version 5.0 in 1998", taking into account "all the feedback we get between now and Thanksgiving 1998." Third, Miller also wanted to write "a proposal (to ACM, IEEE, NSF,…) for us three to get together again in Dec 1998." Last, after that December meeting, they would "start [suggesting] the changes to the powers that be." Apparently, after a week of vacation, Miller was willing to go on a little longer "trying to please everyone over and over."

Rogerson responded two days later (December 29)—with a copy to Miller. Though giving no indication that he had seen Miller's message, he too suggested that Gotterbarn "send an email explaining that we are now in Version 4 and that their comments will be fed into the ongoing maintenance and development of the code and its associated accessories (teaching packs and so on)." Apparently, Rogerson did not anticipate Version 5, but he was sure that the executive committee "must keep all this feedback and eventually write up the findings/outcomes in a paper for CACM or ???"[71]

10.8 What next?

That was a good ending to 1997. Would 1998 be a happy new year? The executive committee could not know. They did not even know whether SEEPP still existed or had, upon delivery of Version 4, automatically dissolved into so many individuals again. Indeed, they did not even know whether the Steering Committee still existed. It had, after all, only five months ago threatened to go out of existence at the end of 1997. It gave no indication of life. The transmittal of Version 4 had not been acknowledged. There had not even been a "thank you" from Cabrara or Frailey. If the Steering Committee no longer existed, who would guide the code through ACM and IEEE-CS approval? New problems to replace the old.

10. Appendix:

SOFTWARE ENGINEERING CODE OF ETHICS AND PROFESSIONAL PRACTICE

(version 4) as recommended by the
IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices

Short Version

PREAMBLE

The short version of the code gives the aspirational elements 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 always act consistent with the public interest.

2 CLIENT OR EMPLOYER - Software engineers shall act in the best interests of their client or employer consistent with the public interest.

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

4 JUDGEMENT - Software engineers shall maintain integrity and independence in their professional judgement.

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

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.

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. Those who contribute, by direct participation or by teaching, to the analysis, specification, design, development, maintenance and testing of software systems 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 this power 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 engineers' humanity, special care owed to people affected by their work, and 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, and 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 which 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 judgement 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 speculate on 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 judgements 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 which 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. Since it 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 always act consistent 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, has passed appropriate tests, and does not diminish quality of life or harm the environment.
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. Co-operate in efforts to address matters of grave public concern caused by software or related documents.
1.06. Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents.
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. Volunteer professional skills to good causes and contribute to public education concerning the discipline.

Principle 2 CLIENT OR EMPLOYER Software engineers shall act in the best interests of their client or 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.
2.06. Identify, document, 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 legislation, 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 are of 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, the client, 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. Work to identify, define and address ethical, economic, cultural, legal, and environmental issues related to work projects.
3.04. Ensure that they are qualified, by an appropriate combination of education, planned education, and experience, for any project on which they work or propose to work.
3.05. Ensure an appropriate methodology for any project on which they work or propose to work.
3.06. Work to follow industry standards when available that are most appropriate for the task at hand, departing from these only when 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 estimates of cost, scheduling, personnel, and outcomes on any project on which they work or propose to work and provide a risk 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 only in ways properly authorized.
3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrences
3.15 Treat software maintenance with the same professionalism as new development.

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

4.01. Temper all technical judgements by the need to support and maintain human values.
4.02 Only endorse documents 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 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 administration of software development. 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 other confidential information.
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 estimates of cost, scheduling, personnel, and outcome on any project on which they work or propose to work, and provide a risk 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 better 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.
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 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 disreputable businesses and organizations.
6.11. Consider 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, ineffective, or dangerous.
6.13. Report significant violations of this Code to appropriate authorities when it is clear that consultation about the problems is impossible, ineffective 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 opinion, concern, or complaint 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. 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 literate 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 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 influence others to undertake any action which involves a breach of this Code.
8.08. Consider 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:
Chair: Donald Gotterbarn;
Executive Committee: 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] Gotterbarn\Version 3\Comment\Unger-4 (addressed to Gotterbarn only). The "4" after "Unger" indicates that this was the fourth "ballot" in Gotterbarn's tally (and perhaps the fourth to arrive). One of the first three seems to have been Notkin's (no number); LaRue's (below) seems to have come third. Whose was the other? Is one lost? Or was it just a ballot without any comment? Neither the archive nor Gotterbarn can tell.

[2] Unger, who holds a BS (Brooklyn Polytechnic, 1952), MS (MIT, 1953) and ScD (MIT, 1957), all in electrical engineering, had worked at Bell Labs (1957-61) before becoming a professor. As part of a re-organization there, he was asked to head a group to develop a compiler for the first electronic telephone switching system (ESS). He agreed to do that for a limited time. He was in charge of the compiler group for two years. Later he did theoretical work on software (e.g., a program for a syntax analyzer). He is now a professor in the Computer Science Department of Columbia University and also a professor in the Electrical Engineering Department. Unger dislikes the term "computer science": "Science is about understanding nature. Engineering is about making artifacts. Software is an artifact. Computer science is about making computers and software. That's engineering, as far I am concerned." Interview of Unger, November 13, 2002. Though never involved in SEEPP, he had been in contact with Gotterbarn for some time. His name first appears in Gotterbarn\SEEP 1994-96\PCVOL3 (file date 6-8-95) in an unsigned note (apparently sent to Gotterbarn by some third person):

> Steve Unger is on the new IEEE International Ethics Committee and with others we are helping Steve implement a mechanism to help IEEE members (later we hope to extend to others working in EE/CS) in ethics matters. The support of your task force is desired. (We of course need to tell you about it. By CC to Steve I'm asking him to do so.
>
> Apart from that, I would like to be involved in SEEPP in some way. Thanks.

The official title of the committee in question is "IEEE Ethics Committee". The parent organization of IEEE-USA is just IEEE (though most members I know tend to refer to IEEE-USA as "IEEE" and the world-wide parent as "IEEE International"—when they bother to distinguish the two).

[3] Stephen Unger, *Controlling Technology* (John Wiley and Sons: New York, 1994). Among his important articles then were: "Role of Engineering Societies in Controlling Hazardous Technology", *Journal of Professional Issues in Engineering* 112 (July 1986): 151-157; "Would Helping Ethical Professionals Get Professional Societies in Trouble?" *Conference Record – Electro*, 1987 (5. 4. 1-5. 4. 6); and "BART Case: Ethics and the Employed Engineer", *Communications Society* 12 (May 1974): 11-13. He had, of course, also published a great many technical papers.

[4] The reason for the substitution of the 1990 code for the 1987 one is not clear, but it seems to be a product of complex relations between IEEE-USA and its parent organization ("IEEE").

[5] These were the Ethical Guidelines Walter Elden's group was working on (9.1).

[6] The phrase "fair and truthful" was my rendering of the standard language of many engineering codes, "objective and truthful". I had substituted "fair" for "objective" because of post-modern concerns about the possibility of objectivity. Unger's suggestion may be an instance of a tendency all writers notice. Change one word in a well-known phrase and it becomes much easier to change others. Perhaps if I had not substituted "fair" for "objective", Unger would not have suggested the double negative, "avoid deception". On a better day, he might have suggested something more positive, such as "candid" for "truthful". Anyone who writes a code will, I think, soon be surprised at what gets read in or read out. Gotterbarn still thinks that my "fair and truthful" does not "not clearly eliminate the possibility of deception by omission". Gotterbarn Chapter9.cmtd (September 29, 2004). I thought "fair" eliminated deception by omission.

[7] Gotterbarn\Version 3\Survey Comments\LARUE-3.

[8] The six Uncertains were: 1.10 ("respect privacy"), 1.11 ("derive data from legal sourses", 1.13 ("identify ethical, economic…issues"), 2.01 ("disclose dangers"), 6.07 ("remuneration appropriate to qualifications"), and 8.08 ("violations of the code").

[9] The two Opposed were: 2.07 ("not put self-interest") and 3.07 ("refuse to participate").

[10] In Version 1, 2.01 had read: "Disclose to appropriate persons any danger that the software or related documents on which they work may pose to the user, a third party, or the environment."

[11] In Version 1, this clause was 2.06: "Not put self-interest, the interest of an employer, or the interest of a client or customer ahead of the public's interest." Version 3 had dropped "customer" (presumably because it is redundant) and added "the interests of the user" (presumably because the user's interest is not identical with the public interest). I regard this change as an improvement—and clearly not the cause of the opposition of our Omaha respondent.

[12] Perhaps the Omaha respondent was proposing a return to the approach taken in the IEEE's first code of ethics (the 1914 code of ethics of what was then the American Institute of Electrical Engineering). Paragraph A.2 read:

> It is the duty of the engineer to satisfy himself to the best of his ability that the enterprises with which he becomes identified are of legitimate character. If after becoming associated with an enterprise he finds it to be of questionable character, he should sever his connection with it as soon as practicable.

Engineers seem eventually to have decided that such a pre-qualification clause did not provide enough protection for the public.

[13] The respondent from Omaha objects to 3.07 that "'refuse' is too strong" and to 6.07 that he does not see how "appropriate remuneration [is to be] determined".

[14] Gotterbarn\Version 3\Survey Comments\CARTMEL. The author of this email was then the principal in Cartmell Technologies of Kingston, New York, a consulting firm advising on assistive technologies. http://www.ulster.net/~dcartm.

[15] Clause 1.07 (Version 3.1) read: "Ensure realistic estimates of cost, scheduling, personnel, and outcome on any project on which they work or propose to work and provide a risk assessment of these estimates."

[16] Gotterbarn\Version 3\Survey Comments\TAYLOR. This respondent has a BS (1989), an MS (1991), and a Ph.D. (1994), all in Computer Science and all from University of North Carolina-Chapel Hill. His dissertation was titled "The Nanomanipulator: A Virtual-Reality Interface to a Scanning Tunneling Microscope". Since 1994, as an NSF CISE Postdoctoral Research Associate (and with additional obtained funding), he has directed the nanoManipulator project. Research interests include virtual environments, man-machine interaction, scientific visualization and distributed systems. http://www.cs.unc.edu/ ~taylorr/taylorr_CV.html.

[17] Email (Gotterbarn to Davis), February 22, 2004.

[18] November 13, 1997.

[19] "Nov 3 The steering committee has asked me to get comments from industry". Gotterbarn\History of SE Code\History expanded.

[20] E971118 and E971120.

[21] Gotterbarn\Version 3\Survey Comments\LANE40. The number "40" seems here to indicate the place of Lane's response in some final reckoning, not its place in the temporal order of responses. So, for example, LINDLEY29 arrived two weeks after LANE40.

[22] The TI respondent must have been thinking of large systems designed for sophisticated clients, not shrink-wrapped software sold to ordinary consumers.

[23] Clause 1.12 (Version 3.1) read: "Whenever appropriate, delete outdated or flawed data". Clause 2.02 read: "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."

[24] By December 1, 1997, there were at least 65 ballots. A few others seem to have come in too late to be counted. I say "At least 65" because that is the highest number of votes on any line of the table saved in Gotterbarn\Version 3\V3 (a file saved on 12-3-97). Is 65 a good response or a bad? In proportion to the hundreds of thousands of software engineers who could have responded, 65 seems almost no response at all. But, given the

quality of the responses, and the small number of respondents in any step of the process so far, 65 looks pretty good.

[25] Email (Gotterbarn to Davis), February 22, 2004 (departmental support). Gotterbarn\Version 3\Survey Comments\ALLCMNTS. This is identical with: Survey Comments\CMTS20NOV.

[26] This is a good idea for dealing with a persistent complaint resting on an undefended prejudice. But it is not as radical or as ACMish as Gotterbarn makes it sound. On the one hand, even Version 1's Rules were designed so that (minus the "In particular…") they could stand alone to form a short code. After all, I had before one major engineering codes (ABET's) that had a short and a long version related in something like this way. The executive committee did not need to change the wording of any Principle. On the other hand, the ACM, having originally printed its document with the code of ethics first and the guidelines after (with each set of the guidelines preceded by the appropriate provision of the code) was by 1997 printing just the guidelines. ACM seems to have treated the short version as a mere inconvenience necessary for approval. By 1997, what Gotterbarn was proposing was not (or, at least, no longer) the "ACM model".

[27] IIT Archive, 1997, New Folder (November 21, 1997).

[28] IIT Archive, 1997, New Folder (November 21, 1997).

[29] Gotterbarn\SEEP1994-65\OPGUIDE.

[30] 4.3.5.c ("This category is provided to allow for ballot returns from members who do not wish to review documents because of conflict of interest, lack of expertise, or other reasons. A reason shall be given for this vote; otherwise, the ballot shall be classified as 'no response.'")

[31] Gotterbarn\Version 3\Survey comments\STPAUL.

[32] Gotterbarn, email (March 2, 2004) added that smaller companies seldom have such adoption problems. His Google search using "Software Engineering Code of Ethics" even revealed small companies that use adoption of the code in their advertising.

[33] Gotterbarn\Version 3\ALLCMNTS (file date 12/3/1997). This seems to be the same as: Gotterbarn\Version 3\Survey Comments\ALLCMNTS (file date 12/1/1997). The earlier document is actually one third longer, but there are several anomalies (such as four pages in which words are printed one letter per line) that make comparing length harder.

[34] When there was no vote at all on a particular clause, the blank was ignored in all calculations.

[35] Gotterbarn\Version 3\VER3mod.

[36] The twenty clauses 3b set in bold for inspection were: 1.04, 1.05, 1.06, 1.08, 1.12, 1.15, 2.02, 2.05, 2.07, 2.08, 3.05, 4.02, 4.06, 4.09, 6.03, 6.05, 6.07, 6.13, 7.02, and 7.08.

[37] So, for example, the list of names at the end of the code ("members") now include Weil and me, although there is no indication that it has been changed.

[38] Gotterbarn, email (March 3, 2004).

[39] These comments are now lost. Gotterbarn recalls that "some commented briefly on several of the clauses, …describing why they had voted negative or positive, advocating that a clause be made even stronger, etc." But, from the summary of comments, it seems to me that the faxed ballots could not have contained much beyond the vote itself—or there would be comments quoted in the summary that do not correspond to emails saved in Gotterbarn's archive. I have been unable to identify any. Gotterbarn, email (March 4, 2004) and (March 8, 2004 attachment).

[40] Email (Miller to Davis), March 15, 2004.

[41] Gotterbarn, email (March 4, 2004). Miller, email (March 15, 2004).

[42] Gotterbarn, email (March 4, 2004).

[43] Gotterbarn, email (March 3, 2004). Compare Miller (email, March 15, 2004):

> Many academics love to debate (myself included), but often that can lead to acrimony. In this case, I found it exhilarating to be in an intense, intellectual exchange while knowing that each of us was working towards the same goal—a better code. I think we trusted and respected each other enough to say exactly what we thought, and I never felt attacked or even annoyed. We could get exasperated sometimes, but I remember us being in good humor most of the time. (Don and Simon "ganged up" on me once because I didn't like a word or spelling, and they assured me it was good English, from the United Kingdom, and that I was being a contrary Yank, or something like that.)

[44] Gotterbarn on this point (Email, March 3, 2004):

[45] While 7.08, with one of the weakest endorsements (85%), was marked for revision or deletion, 2.03 was not. With almost 97% in favor of it, it should have been immune to change, much less deletion. At the least, such deletions should have been noted. Gotterbarn\Version 3\V3 Survey Votes.

[46] Version 4 was 2706 words compared to Version 3's 3407 (and Version 1's 1517). It is interesting to note that Version 4 is about the same length as Version 2.1 (2759). It is 3.0 (3407 words) that is the outlier among the executive committee's codes—and, indeed, among major codes from engineering and computer science. The complete ACM code is

2837 words; the NSPE's, 2218; and the ABET Guidelines, 2666. Did Gotterbarn realize that Version 3 was unusually long? He certainly knew that Version 4 was much shorter. See Gotterbarn\Version 4\MODS ("A brief summary of the general direction of the modifications on version 3 to make 4"), a rough draft never completed (file date 12/16/1997): "The Code was reduced in size by 30%...."—There seem to be at least three ways to calculate the reduction from 3.0 to 4.0 so that it is 30% (as opposed to the 25% I calculated): 1) treat Version 4 rather than Version 3 as the base (though that would yield a 33% reduction); 2) treat only the first digit of the percentage as a significant number (and rounding up to 30% from 25.6%); or 3) combining 1 and 2 (rounding down from 33%).

[47] Gotterbarn\Version 4\S12), December 12, 1997.

[48] Why had I responded to this call for comment when I had not responded to others Gotterbarn had issued? In part, I responded because I could easily do so. The request had arrived when classes were over, all papers graded, and the final exam ready but not yet given. I had a relatively clean desk. (Had the request come a week earlier, I would not have been able to respond in time, since I was then out of the country—and offline.) But, in part too, I responded because I thought it was time to give some advice on drafting. I was hinting that the executive committee would do well to choose me as the editor I had suggested. Their choosing me for the job did not seem as much of a stretch as it would have six months earlier. By then, Gotterbarn knew much of the story of Version 1 because I had asked him to comment on a first draft of a little paper eventually published as: Michael Davis, "Writing a Code of Ethics by E-Mail: Adventures with Software Engineers", *Science Communication* 21 (June 2000): 392-405. (I thanked Gotterbarn for his comments in a postscript to the December 9 fax.)

[49] I also suggested that 4.04 should not begin "Do not engage" because the "shall" in the introductory "In particular" phrase above controls: "one modal is enough." (The final Version 4 corrected this error.) From my perspective, and perhaps from the executive committee's, we were involved with "debugging" a pretty "buggy" code.

[50] I also took exception to the language of this Principle: "'within confines of'. Dead metaphor. Presumably, it means the same as the shorter (and more common) 'consistent with'." The final Version 4 was revised accordingly.

[51] Gotterbarn probably did not treat my comments in this way because a three-page fax was not easily forwarded to the list. He would either have had to retype fax or scan it (and, in 1997, scanners were still not nearly as common, reliable, or easy to use as they would be even two years later). Gotterbarn must, however, have passed my comments on to Rogerson and Miller (in some form or other). Miller and Rogerson (in their responses to Fairweather's comments below) each refer to my comments once (though both call me "M Davies", indicating a single, but mangled, source). The references are to different parts of my comments.

[52] Gotterbarn\Version 4\KEITH.

[53] I had a different objection to Principle 3, specifically to its "products and related modifications": "Modifications are generally modifications of something. If of products, they are products. If of something else, what? Frankly, 'modifications' just strikes me as confusing." Even Version 5.2 retained this unfortunate expression.

[54] In the next version of the code (December 18, 1997), the short version had two paragraphs instead of one, the paragraph coming first doing as Fairweather had suggested. Rogerson's okay was not always so potent, however. For example, having noted the change from "employee" to "software engineer" in 5.02 and other clauses under Principle 5, Fairweather pointed out that a software engineer "may be leading a team that includes people other than software engineers (for example, administrative support)." The good software engineer-manager "should not punish +any+ employee for expressing ethical concerns about a project, or ask any employee to do anything inconsistent with this Code, [and so on]." Rogerson seemed to concede the point, "Interesting—suggest we change 'a software engineer' to 'anyone'." Miller did not object. Yet, the next version—and all its successors—preserved "software engineer". Apparently, sometimes Gotterbarn's vote was all that mattered.

[55] "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." Philosophers may wonder at the similarity between Miller's language and a famous aphorism in Kant's *Critique of Pure Reason*: "Thoughts without contents are empty, intuitions without concepts are blind."

[56] "Gotterbarn\History of SE code\History expanded" reports that the web posting occurred two days later: "Dec 18 version 4 place on SEERI web site announcing that it was recommended to the two societies."

[57] The expression "being sent" is, perhaps, a bit misleading. The suggestion is that the document is even now on its way to the Steering Committee. In fact, from dating on the files of supporting documents, it seems likely that Gotterbarn did not actually email the "final version" to the Steering Committee until the next day, December 16 (after his grades had been turned in). December 16 is also the date given in Gotterbarn's own chronology: Gotterbarn\History of SE code\history expanded ("Version 4 to both societies for approval- December 16 1997"). A related technicality: By Gotterbarn's usual measure, the version sent the Steering Committee (what the email to the task force called "Version 4.DONE") should have been "Version 4.1", since its immediate predecessor was Version 4.0 (December 7, 1997) and that Version 4 differed from this Version 4 in many ways small and large. In fact, "Version 4. DONE" (December 15, 1997) would thereafter be Version 4. This *is* confusing. So, wherever there is a risk of confusion, I will date the version of 4 in question or otherwise make clear which I intend. (Gotterbarn explained his notation this way: "Even though there is a big difference between the 4 of December 7 and the 4 of December 15, we did not bother to assign different version numbers to the paper copies because we did not consider the submission

to the task force to be a release, only a step in drafting." Email, Gotterbarn to Davis, March 8, 2004.)

[58] Because Rogerson was writing from England, the timestamp for this message is 2:30 PM (that is, 9:30 AM CST).

[59] Late that afternoon (December 16, 1997), Ron Prinzivalli wrote the listserv, "What is this?" Was he asking about Version 4.DONE or the parties in Rotterdam and Washington? We cannot tell. No one seems to have responded to his cry for clarification—and he was already dead when I called for an interview (September 2003).

[60] December 17, 1997.

[61] Gotterbarn\Steering Committee\Recommendation of 4\REC (December 16, 1997). The letter begins (after the date) with the salutation: "To the Joint IEEE-CS/ACM Steering Committee on the Professionalization of Software Engineering and Executive Committees of the ACM and the IEEE-CS." Because the file is not itself an email, we cannot tell to whom this message was sent or when. But it seems reasonable to suppose that Gotterbarn would have sent it to Cabrera, with a copy to Frailey, as he regularly did when dealing with the Steering Committee, and that he would have sent the letter to the other Committee members through the presiding officer (rather than directly). The only confusion seems to concern to whom the recommendation was sent. "Gotterbarn\History of SE code\History expanded" mentions only the two societies (omitting the Steering Committee). This seems a slip. The Steering Committee's chair and co-chair never denied receiving the December 16, 1997, letter of transmittal. The importance of these details will become obvious in the next chapter.

[62] For more detail, including the exact working group involved and what it reported, see 8.7 (and endnote).

[63] The archives contain no record of this correspondence.

[64] These requests are not in the archive.

[65] Gotterbarn\Version 3\Feedback on 3\Wentworth (December 22, 1997).

[66] Gotterbarn\Version 3\Survey comments\Siemenscommnt.

[67] The language of these clauses does not matter here. For the exact wording of these clauses, see 9. Appendix.

[68] Version 3.0's 6.10 ("Obey all laws governing their work, insofar as such obedience is consistent with the public health, safety, and welfare ") had become Version 4's 6.04 ("Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest"). Would this 6.04 have satisfied the German? What of the ancestor of all these, Version 1's 5.10 ("Obey all laws governing

their work, insofar as such obedience is consistent with the public health, safety, and welfare")?

[69] The original of this email has not survived, but we have it complete both in Miller's response (Gotterbarn\Version 3\ Survey Comments\AKM) and Rogerson's (Gotterbarn\Version 3\Survey Comments\ASI).

[70] Gotterbarn\Version 3\ Survey Comments\AKM.

[71] Gotterbarn\Version 3\Survey Comments\ASI.