

Chapter 7: Winter Whirlwind, Version 2

I bear orders from the Captain:
“Get you ready quick and soon,
“We must all be together
“At the rising of the moon.”
—Irish folk song

7.1 Statement on codes of ethics

On January 3, 1997, Donald Gotterbarn and his wife Shirlee arrived at London’s Heathrow Airport. Not sure what to expect of English weather from January to July, they had packed one of everything, filling four large suitcases and two carry-on bags. They took a bus to St. Pancras Station and there boarded the Northern Express, struggling with their luggage so much that the conductor called ahead to Leicester to be sure a luggage handler would be ready to help them off the train. An hour later, they were stepping onto the platform at Leicester, a city of a quarter million people, once a center for manufacture of knitwear, hosiery, and shoes, now the home of many small high-tech companies and two young universities. The station is a short walk from the nineteenth-century City Centre and from De Montfort University’s Leicester Campus. Simon Rogerson was waiting on the platform (along with the promised luggage handler). Having greeted them, Rogerson led the way to his car, loaded as much of their luggage as he could while leaving room for them, and put the remainder in a taxi. The taxi following, he drove to the terrace house, a half mile from campus, where they were to live for the next six months. A light snow was falling. The city looked like a postcard.

On Monday (January 6), Gotterbarn arrived at the suite assigned the Centre for Computing and Social Responsibility (CCSR) on the tenth floor of the James Went 2. (Went was a split-level high-rise. One side of the split was Went 2; the other side, a half-storey down, was Went 1.) A gray, rundown, and undistinguished building, marked for demolition early in the next decade, Went contained both offices and classrooms—as well as a snack shop and other amenities. CCSR’s suite consisted of one large open square with desks for a Research Associate (Ben Fairweather), three students, and the “webmaster” (a technician who maintained the server for CCSR’s website). Windows along one side made the open square bright. Having shown Gotterbarn around, Rogerson led him to the second floor of Went 1 where his office, formerly a storage area, was waiting. A (more or less) triangular room with a desk “3-4 feet long”, a computer, a book case, and two chairs, it was just large enough for two people of modest proportions to meet. But it was not without advantages. It had a picture window the length of its longest wall, was convenient to toilets and elevators, and had one other advantage, though Gotterbarn would not appreciate that one for several weeks: As far as DMU was concerned, the office was still a storage closet. When Rogerson left his faculty office just down the hall to meet with Gotterbarn, he (in effect) disappeared from DMU. No one would look for him in a storage closet. Indeed, no one but an initiate would look for Gotterbarn there either. The two of them could work in Gotterbarn’s office even for long periods without interruption.¹

The preliminaries over, Gotterbarn was soon checking his mail on the DMU (desk-top) computer.² A message he had been waiting for—from Cabrera and Frailey— was there.

Addressed to “Software Engineering Profession Steering Committee, Task Force Chairs, and Ex-Officio Members of the Steering Committee”, it formally announced what Gotterbarn had agreed to two weeks before:

In order to expedite the remaining activities assigned to this task force, we have asked Donald Gotterbarn to assume full chairmanship. The task force has the following activities to complete by February 14:

- complete a draft code of ethics and professional conduct that has its roots in the codes currently in use by other engineering professions as well as the codes of the IEEE and the ACM
- document the architecture of this code so that reviewers can understand the components, which parts are derived directly from other codes, and which parts are specific to software engineering
- document the rationale for the new parts of the code (those not derived from existing codes or which depart from existing codes in substantive ways)

Other activities for the remainder of the year will be outlined in a plan that Don will distribute next week.

We understand that for a variety of good reasons, many members of this task force have been unable to participate at the level originally planned. We specifically authorize Don to drive the task force to completion, making whatever changes in membership or structures are deemed necessary to accomplish this task.

It was now official. Gotterbarn was no longer tied to an IEEE co-chair or to the IEEE standard-writing procedures. He was free to act. If SEEPP failed now, he alone would be responsible.

Gotterbarn forwarded the Steering Committee’s email to his working group’s listserv, adding his own covering message. The message’s first paragraph announced “some changes in the SEEPP task force” and directed readers to the forwarded message from Cabrera and Frailey. The second paragraph (in part) explained the new listserv:

In order to address these tasks [the three listed in the Cabrera-Frailey message] I have formed a combined working group [sic] collected under a single email list called prfcmp-1. This explains why several of you have received email saying you were added to this list.

Gotterbarn then promised to send “a copy of the plan” the next day and “hope[d]” to send “a draft statement of the roles and functions of professional codes of ethics...by Thursday [January 9].” Last, he informed the reorganized “working group” that he was “now working at” CCSR, that he had a new email address (dgot@DMU.AC.UK), and that the listserv should not be used for a message directed to an individual (since many more than that individual would see it).

Gotterbarn did not make much of it at the time and many, perhaps most, of those active in SEEPP’s work may not have noticed (or cared), but Gotterbarn had just reorganized SEEPP. Everyone who was not already in Gotterbarn’s working group was now in it. Though Gotterbarn said nothing about the other working groups, a careful reader might have noticed that the name

given the new listserv implied that Gotterbarn's working group had taken over: SEEPP (the task force) had become the "*Professional Competence Standards* Task Force". The members of the other working groups had all been incorporated into one entity which (apparently) Gotterbarn now indifferently called "working group" or "task force". Gotterbarn had abandoned "divide and conquer". SEEPP now had the unitary organization of Mechler's SEEPP/E. (Gotterbarn would not again mention the other working groups, except once—in 8.1—to declare them dead.)³ He had, however, not yet found a replacement for SEEPP itself—at its best) a small group of advisors with whom he could meet face to face several times a year. Within a few days, he would.

The next day, Gotterbarn distributed Version 1 through the listserv. The covering note began "Dear task force member" (as would all subsequent messages he sent through the listserv). The note asked for help with "[annotating] the code with references to similar or identical imperatives in other codes"; it promised "a draft of the codes of ethics statement" the next day (Wednesday). There was no mention of "the plan" promised the day before (and, in fact, Gotterbarn had decided not to distribute it until the Steering Committee approved).⁴ Gotterbarn gave no indication of when volunteers should turn in their annotations or how the work was to be divided among them. (They would, of course, know that they had to get their work in before *Gotterbarn's* deadline, but not the last date before that on which what they did would still be useful.) Perhaps Gotterbarn expected to set deadlines and divide the work once task force members volunteered. If that was what he expected, he was to be both disappointed and surprised.

On January 8 (as promised), Gotterbarn distributed the draft "code of ethics statement". Its covering note said it was "a brief document describing the roles and functions of codes of ethics in an emerging profession" that he wanted "to modify...so that it can provide some guidelines for us as we revise the draft code of ethics and determine how next to proceed". Gotterbarn concluded by inviting "comments on its content and style". Again he gave no deadline for response. The statement, about three single-spaced pages, seems intended as a summary of everything then known about the functions of codes of ethics (or, at least, of codes of ethics in "emerging professions"). But the eight references at the end suggest something much narrower. Three of the eight, including two to Gotterbarn's own previous work, are explicitly about codes of ethics in software engineering.⁵ Three more are computing-related—Anderson's paper on the ACM code, Jacques Berleur's on the code of ethics of the International Federation of Information Processing (IFIP), and Johnson's *Computer Ethics*. The last two references are to textbooks in *engineering* ethics—Johnson's anthology (1990) and *Ethics in Engineering* by Mike Martin and Roland Schinzinger (1989).⁶ These two represent the best texts in engineering ethics of the 1980s, but neither they nor the rest of the references reveal a wide familiarity with the literature of professional ethics or even an updating of Gotterbarn's earlier reading in engineering ethics.

Gotterbarn was no doubt doing the best he could in the few days the Steering Committee's schedule gave him. The Committee had, in fact, allowed much less time than a mere counting of days on the calendar would suggest. During his first few weeks in England, some time had to go to setting up house, learning his way around campus, and identifying resources he would need to draw on later. His sabbatical leave had two large projects in addition to completing revision of the code of ethics. He was to "work up the concept of Software Development Impact Statements" and develop a method for "ethical\professional risk analysis".

He was also making one round trip a week to the United States to teach a ten-week course at the National Security Agency (in a Washington, DC suburb), giving lectures in various classes at DMU, and preparing papers for conferences. Gotterbarn even lost a day or two fighting a virus that infected his computer.⁷ Like all those in SEEPP (and its working groups), Gotterbarn was always an unpaid volunteer, his work on the code having to compete with activities that paid the bills. That was true even during a sabbatical year.

Gotterbarn's statement began by noting a "negative standpoint" on the function of codes of professional ethics as well as a positive. The negative understands codes as "merely a self serving attempt to generate a positive public image" or "to establish a moral minimum" rather than a "complete" guide to professional conduct. While granting that these "negative judgments" must be kept in mind when preparing a code, the writing of a code should (according to Gotterbarn) focus on the positive functions. The positive functions can be achieved using one or more of three sorts of standard: 1) those that apply to everyone in virtue of a common humanity (honesty, for example); 2) those that apply by virtue of one's special role (or skill) as a professional (for example, the obligation to serve society); and 3) those that apply because of the special structure of a particular profession. Gotterbarn offered no example of standards of the third sort (in this statement or in the earlier work he referenced) but he seemed to regard such standards as (more or less) deducible from the particular profession's powers, skills, and role (just as service to society is deducible from the special role of professions as such). Gotterbarn also did not indicate whether these three sorts of standard correspond to what he elsewhere distinguished as: code of ethics; code of conduct; and code of practice (for example, in his email of October 8, 1996, to Frailey).⁸ Practical (rather than scholarly) considerations seem to control what he says here, the most important of which would be satisfying the Steering Committee.

The core of Gotterbarn's statement is a list (and explanation) of the (positive) functions of codes under six headings: "INSPIRATION", "GUIDANCE", "EDUCATION", "SUPPORT FOR POSITIVE ACTION", "DETERRENCE AND DISCIPLINE", and "ENHANCE PROFESSION'S PUBLIC IMAGE". The statement seems designed to lay a foundation not only for defending the final code (whatever its form) but for rewriting Version 1 to make it more like the ACM's code (which explicitly sorted standards according to something like Gotterbarn's three-part plan—with enforcement adding a forth part). The statement also seems (if less obviously) a justification for not connecting the code with licensing. The statement's concluding paragraphs explicitly limits the software engineering code of ethics to the first four functions. The code would *not* be designed for deterrence or discipline (as a code designed for licensing *would* be) because "[at] this stage..., the disciplinary function is being taken over by the law." The code would also not be designed to improve the profession's image because "[in] most contemporary codes, the attempt to keep a perfect public image at the expense of quality development has been abandoned." A code designed neither for disciplining errant professionals nor for improving the profession's public image is unlikely to have much use in licensing.

By 1997, the most important text in engineering ethics included one function not on Gotterbarn's list, one especially important to an "emerging profession" and the very one Mechler had stressed in his exchange with Frailey and Shaw: setting a higher standard than law, market, and morality would otherwise require.⁹ The higher standard (a requirement, not an "inspiration") should if followed, should improve the profession's image (that is, give it a reputation for doing work of a quality higher than law, market, and morality ask of others). Once a profession had such a reputation, government might well want the service the profession provided to be

provided only by members of the profession (as a way of assuring consumers or public a certain level of quality). Might we then wonder why Mechler did not suggest that Gotterbarn add that function of professional codes?¹⁰ And, since the textbook in question lists me as authority for that function of codes, we might also wonder why I also said nothing at the time. Understanding why neither Mechler nor I said anything at the time may help us understand why no one else did either.

One of my reasons for not responding is embarrassingly pedestrian: I did not immediately receive the email. In those days, I only received email at the office. Just after New Years day, I had outpatient surgery on the big toe of my right foot (to fuse two joints). Though not major surgery, it did mean I had to use crutches for six weeks and could not drive. So, that January, I worked at home until the end of winter break (the Tuesday after Martin Luther King Day). Only then did I take a cab to my office, meet my class, and check my mail. By that date (January 21), Gotterbarn's statement was almost two weeks old, stale mail that had to compete for my attention first with more recent mail and then with all the demands of a new semester.¹¹ Without a deadline, Gotterbarn's request for comment had a low priority. Of course, I did put his email in my (paper) file, intending to respond. Within a few days, however, that intention drowned in the new wave of emails Gotterbarn was already preparing. Late January was a bad time to get much of a response from me—and perhaps from other academics.

That was, however, not the only reason I did not respond. I was back in my fly-on-the-wall mode, intrigued by what Gotterbarn was trying to do and wanting to see what would happen. I wanted there to be a code of ethics for software engineers. Without the code, there would not be much interest in the process of writing the code (or, rather, trying to write it). While I had a fatherly interest in Version 1, it was the interest a father takes in an adult child who should be able to prosper on her own. Whatever Gotterbarn's errors, they were, it seemed to me, not his alone. They were the common wisdom of several decades, and would no doubt continue to influence thought about codes for several decades more. There was then no pressing need for me to say again what I had already said. There should be other opportunities; writing the code had been a slow process. There was, it seemed to me, no reason to expect that to change any time soon. Had Gotterbarn personally sought my advice (as Mechler had), I probably would have responded quickly, but I had no reason to respond quickly to the general call for comment. There did not seem to be any deadline (except the date binding Gotterbarn, February 14, that, residing in an earlier email, I did not recall); and there were others, more directly involved, to respond—Mechler, for example.¹²

7.2 Mechler helps

Why didn't Mechler respond? Perhaps he did not receive Gotterbarn's January 8 email. He never mentioned it in later emails and there is no copy in his files. Or, perhaps, he forgot it as soon as he saw what it was. His interest was the code, not theoretical discussions of codes. What we do know is that Mechler took seriously Gotterbarn's request of January 7 for help annotating the code (though he did not tell Gotterbarn). On January 28, 1997, with the deadline for submission of everything to the Steering Committee approaching (February 14), he emailed Norman:

The request from Don, dated 1/7/97, Draft Code, asked for annotation. I am doing a number of other codes but have trouble with the one you sent AICPA Standards Manual because it was extracts. Can you annotate the Code from the manual? Let me know.

Two hours later, Norman responded: “I don’t have the manual any more. I guess I could get it from the University Library again, but I might not be able to get it done in time.” Early the next morning (January 29, 1997), Mechler told Norman to go to the library only if he wanted to—and then, after explaining further why he needed the “exact reference”, complained that the annotation “is taking forever”, asked “Have you heard anything from the list since Don sent his last e-mail?”, and concluded by setting the “end of the week” as his deadline for “sending out the references”. By 11:09 AM (after several more messages back and forth), Norman announced, “Good news. I have the original that I sent you and it has references to sections and paragraph numbers from the AICPA Manual....Apparently, I’m more organized than I thought!” Mechler then suggested that Norman “annotate Dons e-mail of 1/7/97 and put it to the list [that is, send it out through PREFCMP-L]. I will do the same with mine.” Late that afternoon (after an exchange of emails in which each admitted to being “an anarchist”, that is, a person not much given to keeping orderly files), Mechler returned to Norman’s question concerning a deadline: “the only dead line I have is Feb 14th. Did you get the additional schedule mentioned in the forward e-mail from Don?”

The next morning (January 30, 1997), Mechler sent his annotations to the listserv (PREFCMP-L). He cited five “source codes” (those belonging to the IEEE, NSPE, Project Management Institute, ACM, and Institute for Certification of Computer Professionals). He assigned at least one (and up to all five sources) to each of the particular rules. For example, under 1.01, he listed “PMI IIb— ACM 3.4”. Mechler made no attempt to provide a source for the Introduction or for any of the seven general Rules. He prefaced the annotation with a caveat. A reference may mean that the particular provision of the code is either:

Exactly as stated in another code
Paraphrased from other codes
Expanded for SE from other codes
Idea generators from other codes.

So, if readers “decide another reference within the codes annotated is better” or “have additional codes not mentioned”, they should add the reference. But, anyone who disagrees with a reference should email Mechler and “we will discuss it”.

Mindful of Murphy’s best-known “law” (“If anything can go wrong, it will”), Mechler waited a half hour and then emailed Gotterbarn to see whether the annotations had arrived. Mechler also took the opportunity that this email offered to ask two questions he had wanted to ask ever since he received Gotterbarn’s January 7 email:

1. What was the intent of the statement: “we understand that for a variety of reasons, many members of this task force have been unable to participate at the level originally planned.” In Felipe and Dennis e-mail you forward[ed] to us?
2. What is the plan for the rest of the year they requested in the same e-mail. I do not know if you sent anything yet.

Mechler also emailed Norman to see whether he had received the annotated code. About noon, Norman responded that he had received it. Sure that all was well, Mechler went on to other things. But the next morning (Friday, January 31), he received a message from Weil at IIT: “SOS. The Code of Ethics annotated toward other codes was not attached. Instead came the words ‘Unable to print this part’.” Mechler investigated and responded about twenty minutes later:

It is something to do with the list. I am on the list twice using CompuServe and Exchange and both came with warning about MIME. But I received both. Here is another copy. Let me know if you received it.

Having sent this message, he wrote PRFCMP-L: “There appears to be some problem with the SEEPP List sending my annotated copy of the Code. If you didn’t get it, please let me know at emechler@ero.eqt.com.”

Seven minutes later (about twenty-four hours after Mechler first sent his annotations), Gotterbarn emailed the list a message with a table attached (comdes.doc). According to the message, the table not only incorporated Mechler’s annotations but extended them to seven other codes. A half hour later, Mechler responded: “Can not open comdes.doc. Please resend or send as a text file.” Mechler also emailed Norman directly asking whether *he* was able to open Gotterbarn’s file. Norman took a half hour to reply: “My VMS mail cannot read MIME either. However, I can convert it to text with PINE and I will send you a copy.” An hour after that (still January 31), Gotterbarn notified the list: “The attachment seems to have some problems :-)”¹³ While you are reading and commenting on the tome I sent, I will try to get this table straightened out (he said hopefully!)” Gotterbarn did not solve the problem until late Sunday (February 2). Early Monday morning he wrote PRFCMP-L:

Sorry about the problems with the early version of the table. Only one person (with a MAC) was able to read it. To avoid all difficulties with versions or word processors etc. I spent a pleasant ground hog day (a special holiday in the colonies) rebuilding the table.

Tuesday morning (February 4), Mechler had some bad news for Gotterbarn (using PRFCMP-L): “In the second version of code relationships [comdes2.doc] the references do not line up with the correct code. It appears as spaces, tabs, etc. were not included. Nothing lines up under IEEE or PMI. Norman quickly emailed Mechler directly: “Mine seems to be reasonably lined up; at least, readably so! Here it is.” What lined up for Norman did not line up for Mechler. After several more exchanges, Norman realized that the problem was not visible on his computer screen because of the font used. Everything lined up using the default font, Courier, but not using any font having variable spacing. He advised Mechler to print using Courier. “If that doesn’t work for you, let me know and I will get a printout with Courier and fax it to you.” But, Norman added, waiting for the fax might mean waiting till tomorrow because he would have to use his home PC:

I do not have immediate [access] to the right combination of resources right where I am—Our VMS prints to a lineprinter, and the stuff in labs is occupied by hordes of

students who always succeed in breaking things. We have the equipment, but to get a working combination of monitor, disk drive, and printer in the middle of the semester is quite an achievement here!

Norman had figured out what the problem was and had told Mechler—but not the listserv (or Gotterbarn). Norman and Mechler seem to have forgotten that they were working “off the list”—or, perhaps, it did not occur to either of them that Gotterbarn might also be working on the problem. But Gotterbarn was working on it and, on February 5 (Wednesday), he wrote Mechler triumphantly:

The copy of the code you sent me lined up on my system, standard IBM clone. I checked with someone else here and they had no problem either, so I thought maybe something flipped going across the atlantic and I check[ed] with someone else in the states. They didn't have a problem either. Let me know your fax number and I will send you a hard copy of the table.

Mechler sent the fax number, saying nothing about Norman's discovery of the day before and, indeed, without any message at all. Did this unusual curtness mean that Mechler was too busy to chat (or that he had had enough chatting about the attachment)? While we cannot answer that question, we can be pretty sure that the reason Mechler sent the fax number was that Norman's discovery had not solved his problem and Mechler wanted to see Gotterbarn's table. Gotterbarn nonetheless had to put off sending the fax. The Went Building had (he soon learned) only one fax machine, requiring a code and authorization to use. Rogerson, who would have had to authorize Gotterbarn's use, was out of town. So, instead of faxing, Gotterbarn kept working on the problem. By the next morning (February 6), he had remade Norman's discovery (or, rather, confirmed a hypothesis Ben Fairweather, the philosopher, had posed).¹⁴ Gotterbarn first wrote Mechler directly and then informed the listserv, adding new details:

The table is pure ascii text table and must be printed using a mono-space font, something exciting like courier :-). Copy the table onto your word processor, select the table, select a mono-space font which does not cause the table to wrap. If there is nothing under PMI for 1.01 then the table is wrapping. This should work for most systems that I know of.

Norman confirmed a few minutes later (using the listserv): “Yes, I discovered that. Any of the courier fonts works fine with it in MS Word.” Early Friday (February 7), Mechler also confirmed (using the listserv): “Thanks everyone. I finally got a lined up copy of the matrix. Thanks Don.” It had taken exactly one week for three experienced software engineers (two of them still practitioners) to get comdes2.doc to print properly on Mechler's equipment.

7.3 Comparing codes

The table was worth the effort. Below is the first part of it, the imperatives under Rule 1 and their sources (Rule 1 having the fewest sources). (For the entire table, with legend, see 7. Appendix.)

[Insert]

Product	AAES	ABET	ABET-G	ACM	ACM-G	BCS-C	BCS-P	ICCP	ECPD	IEEE	NSPE	PMI
1.0 1				3.4								II.b
1.02	A			3.4								II.b
1.03	C1	B.2 2				20	1.3	2.5		'6'	'II.2.a'	I.b
1.04			1.c.1	3.4								
1.05				1.2					2			II.b
1.06									2			II.b
1.07							2.4		'3'			II.b
1.08		1.c.1		3.4							'II.2.b'	II.b
1.09		1.c.1	2.5									II.b
1.10				1.8, 3.5 1.7	1.8 3.5	2	3.5					I.b
1.11					1.7	2	4.6					
1.12	C2, C4	A.1	1.a, 1.c.1		2.3 3.1	1.1, 1.4	1.7, 8			8		
1.13			4 .j						2			
1.14				1.2								

The rationale for choosing just these twelve codes is not obvious. All are American or British. Why none from the rest of Europe? Why none from Latin American, Australia, or Canada? All the *engineering* codes are American (unless, following the British, we count the BCS's as an engineering code). Why not even one (other) engineering code from Britain (say, the code of the Institute of Electrical Engineers)?

There is another oddity. There are two ACM codes where there should be one. The ACM-G (the commentary on the Code of Ethics and Professional Conduct) includes the entire ACM Code of Ethics (ACM). In this respect, the ABET Guidelines (ABET-G) differ from the ACM-G. Not only is it a document separate from the short ABET Code of Ethics for Engineers; it is also missing some parts of that code (specifically, the Fundamental Principles). Like the NSPE's code of ethics, the ACM code does not seem (properly) divisible into code and independent guidelines.

The codes divide neatly into six for engineering (AAES, ABET, ABET-G, ECPD, IEEE, and NSPE) and six for computing (assuming the Project Management Institute is primarily concerned with managing software projects). But two engineering codes seem odd choices: The AAES code (American Association of Engineering Societies, Model Guide for Professional Conduct), though adopted in 1984, was almost forgotten by the mid-1990s. It is one of those "failed codes" no one would want to copy. The ECPD "code" (Faith of the Engineer, Engineer's Council for Professional Development) is a one-page "creed" adopted in 1948 by what later became ABET. It was never intended as a code of ethics. The ECPD adopted its first (and very successful) code of ethics in 1947 ("successful" insofar as most major engineering societies adopted it within a few years of its publication).¹⁵ ABET let The Faith of the Engineer (the "Faith") die quietly. What did Gotterbarn hope to achieve by citing it? He did not say. The obvious answer is political: give engineering the same number of codes computer science had. But perhaps there is another explanation. Both the "Faith" and AAES code appear (on facing pages) at the back of the Martin and Schinzinger text.¹⁶ Their presence in that book may have misled Gotterbarn about their importance. He expressly cites that text as the source of "[m]ost of the [engineering] codes".¹⁷ There is other evidence for this explanation: *all six* of the engineering codes Gotterbarn cites (and only those) appear in Martin and Schinzinger (1989). Gotterbarn's dependence on that text would also explain why no British code of engineering ethics is cited, though codes of the British Computing Society are.¹⁸ The Martin and Schinzinger text includes no non-American codes.

If this explains Gotterbarn's choice of engineering codes, then we are entitled to draw two conclusions. The first is that neither Gotterbarn himself nor those with whom he was working at CCSR knew much about engineering codes. If engineers (and their philosopher friends) dominated "SEEPP\E", now computer scientists (and their philosopher friends) were dominating the work of the reorganized SEEPP. The second conclusion is that Gotterbarn probably lacked the time to find out what he did not know. He had a tight schedule. The only convenient library (DMU's) lacked a recent American text on engineering ethics. And there were then few, if any, websites displaying codes of ethics.

While I am also tempted to explain the absence of codes of ethics other than those from engineering and computing in the same way, I think there is a simpler alternative. Gotterbarn may have been interpreting the Joint Steering Committee's request (January 6, 1997) that he "document the architecture of this code so that reviewers can understand the components, which

parts are derived directly from other codes, and which parts are specific to software engineering” as *implicitly* asking only about computing and engineering codes. That is not unreasonable for two reasons. First, the Committee’s immediately preceding request (its first) was even more limited (“complete a draft code of ethics and professional conduct that has its roots in codes currently in use *by other engineering professions as well as the codes of the IEEE and the ACM*”).¹⁹ Second, precedents closer to software engineering would naturally carry more weight than precedents more distant. Every clause had precedents close to software engineering. Gotterbarn had no reason to look further. He must not have known that until late in the process, though. Had he known that earlier, he would, presumably, have made clear that volunteers should only look at codes related to engineering and computing (and so saved Norman and Mechler the trouble of recovering notes on the AICPA code).

Whatever reason there is to criticize the choice of codes, the overall message of the table is plain (as even the part reprinted above suggests). Version 1 seems to correspond most closely to computing-related codes: ACM’s and PMI’s. That, no doubt, is—as Gotterbarn points out in his comments—because the imperatives under Rule 1 concern software (with which ACM and PMI are specifically concerned). Yet almost every imperative also has at least one “engineering hit”. Contrary to what Frailey and Shaw argued, Version 1 was not, in substance, all that different from existing codes. It therefore is not likely to be less practical than these. The one way in which Version 1 seems to differ from the others should be an advantage. Version 1 covers more than any other single code does (though not more than all together). Gotterbarn could, it seems, have treated Version 1 as what Cabrera and Frailey had asked to have delivered by February 14, a “draft code of ethics and professional conduct that has its roots in the codes currently in use by other engineering professions as well as the codes of the IEEE and ACM.” Nothing in the table, or in the comments so far received from the working group, required a total rewriting or restructuring (rather than a series of narrowly targeted amendments). What did seem to, if anything did, was the prejudice against Version 1 that the premature criticism of Frailey and Shaw may have generated within the Steering Committee (5.6).

Gotterbarn’s table (the attachment) came with both a one-page single-spaced covering letter and (following it) a two-page single-spaced document bearing the very long title:

Revision of
DRAFT DOCUMENT RELATING
COMPUTING, ENGINEERING, AND SOFTWARE ENGINEERING CODES.
v2 30 Jan 1997
The Architecture of the Draft Software Engineering Code of
Ethics and Its relations to other Codes of Ethics

The authors of this document, “Gotterbarn & Rogerson”, must have begun it at least a few days before January 30, but they must have revised it on January 30—after Mechler’s annotations arrived. The third paragraph says that “the committee” (not “the working group” or “task force”) “examined other codes of ethics including...”²⁰ The list of six items includes W.J. King’s *The Unwritten Laws of Engineering*, the AICPA code, and the PA State Registration Board’s code. The list is plainly Mechler’s, but nonetheless differs from his list of eight in three ways. First, the NSPE code is not there.²¹ The reason for this difference seems to be that the preceding paragraph had already identified the NSPE code’s structure as the model for Version 1. There was no need

to mention it again so soon. Second, the Institute for Certification of Computer Professions (ICCP) Code of Ethics is not there (even though it is one of the twelve in the table). There seems to have been no reason to omit the ICCP code—except that it added little to the variety of sources already listed. Third, the list is not quite in Mechler’s order. Mechler had the IEEE code first and the ACM code fourth. Gotterbarn & Rogerson moved the ACM code to first place. Alphabetical order does not seem to explain this change. If the order were alphabetical (as it is in the table of codes), the AICPAP code would come second, not fourth; the PMI code would not be third; and so on. The reason for putting the ACM code first seems to be political. So far, opposition on the Steering Committee had come from two ACM-appointed members (Frailey and Shaw). Putting the ACM code first here might “help the medicine go down”.

Overall, the statement on “architecture” seems designed to introduce Version 2 (though the first use of “SECEv2” is not until the seventh paragraph). Thus, the first paragraph begins:

This document was developed in response to a request for the IEEE-CS/ACM Joint Steering Committee for The Professionalization of Software Engineering. The architecture of the draft Software Engineering Code of Ethics (SECEv1) has been modified in the light of comments from the software engineering and computer ethics communities.

The fourth paragraph, after reporting that “the committee” had distributed SECEv1 to others for comment, says that “in light of those comments and further work by the committee, the structure of the code has been revised to reflect significant aspects of other engineering and computing codes.” The remaining eight paragraphs describe SECEv2. So, for example, after noting that “[many] codes” such as the British Computer Society’s and the ACM’s include “a preamble describing the roles and functions of the code for the practicing professional”, it indicates that “[the] structure of SECEv1 has been revised in accordance with this model”.

Though most of the document on architecture follow closely Gotterbarn’s early paper on functions (for example, using SECEv1 to illustrate his “three levels”), it does report two surprises. First, the original purpose of constructing the table (“documenting the architecture”) had been to “show which parts [of SECEv1] derived from other [engineering and computing] codes and which parts are unique to software engineering.” That purpose presumed that there are “clear lines of demarcation between engineering, computing, and software engineering.” The table is “not consistent with that presumption.” None of the imperatives is unique to SECEv1, and virtually every imperative has a counterpart both in some engineering code and in some computing code. Mechler had been right. (5.6) Second, there is nonetheless an interesting contrast between engineering codes and computing codes: “At the moment some imperatives are contained mostly in engineering codes (the outer columns of the chart) and [some] imperatives are contained mostly in computer codes (the inner columns of the chart).” Those imperatives sparsely represented in both engineering codes and computer codes concern “management issues” or “very specific...technical issues in software engineering.”

The covering letter (January 31, 1997) preceding the document on code architecture began with thanks to “those who commented on my draft on Functions of Codes of Ethics and who contributed to annotating the code...especially Ed.” This explicit thanking of Mechler seems right, but it does suggest two questions. First, who else contributed to the comments or annotations?²² Second, what in fact was Mechler’s contribution? What did Gotterbarn mean

when (in paragraph 2), he said he had “integrated Ed’s results into the baseline document [the table]”? We can be confident that someone other than Mechler did contribute. Mechler had provided annotations for five codes; the table covered twelve. Norman’s annotations did not arrive in time (and would, in any case, not have fit the engineering-computing plan). Who did the work for the other seven codes? The answer may be: Gotterbarn. His general practice seems to have been to thank explicitly anyone who did any significant work. He had special reason to thank some others if he could, for example, his host, Rogerson. We must therefore suppose Gotterbarn to have analyzed the seven codes more or less by himself.²³ What else must he have done?

We know that Mechler found analyzing five codes to be taxing, even when the work was spread over several weeks. We must then assume that Gotterbarn spread his work over several weeks as well—or, at least, did not do everything in the twenty-four hours between the time he received Mechler’s annotations and the time he sent the table out. But, if Gotterbarn worked on the table for several days or weeks before Mechler’s annotations arrived, he was unlikely to have annotated just the codes Mechler omitted. Gotterbarn did not know what Mechler was doing, or even that Mechler was doing anything, since Mechler did not tell him. Gotterbarn, in turn, did not tell Mechler there was a deadline for the work if Mechler (or other contributors) wanted their work to be of use; and, indeed, there may have been none (apart from February 14). Gotterbarn seems to have worked as fast as he could; the table was just one of several substantial projects soon to be revealed. So, by January 30, he must have had an almost complete table for all twelve codes. He would have included the codes Mechler did (where there was overlap) for the same reason Mechler included them: they were too important to omit (and, except for the ACM, PMI, and ICCP codes) were included in the Martin and Schinzinger text on which Gotterbarn relied). So, the arrival of Mechler’s annotations must have been a complete (and unique) surprise. Their late arrival meant that they could provide no more than a check against Gotterbarn’s own work.²⁴ Had Mechler been a little slower, his work would have been entirely wasted (as Norman’s was). Gotterbarn was no longer moving slowly, waiting for others to do something. Those used to his patient style did not, and probably could not have, anticipated what Gotterbarn did in January—or what he would do over the first two weeks of February.

The rest of Gotterbarn’s January 31 letter was a status report. Paragraph 3 promised to “[revise] the Functions paper in light of your comments”. Paragraph 4 reported that Gotterbarn “[is] fortunate to be at the Centre for Computing and Social Responsibility this semester, which is supplying both intellectual and clerical support of this effort. Paragraph 5 lays out a “tentative [six-step] plan” (without dates): “1. [Submit] a revised code to the steering committee for review and comment, 2. Revise the code in light of the comments, 3. Print the Code in the ACM and IEEE publications with a survey asking for comments and voting on each item”, and so on. The paragraph ends with the promise that a “more detailed plan is forthcoming”.

7.4 The plan to revise

On February 4, 1997, Gotterbarn addressed a sixteen-page (single-spaced) memo, “Dear EC (executive committee)”.²⁵ The “executive committee” had just three members: Gotterbarn, Miller, and Rogerson. The executive committee formalized what had become Gotterbarn’s practice over the two weeks preceding—working closely with Miller and Rogerson, both of whom responded quickly to calls for help and seemed to Gotterbarn to know what they were

doing. Why Gotterbarn thought it necessary to formalize his “kitchen cabinet” is a mystery. As a formal body, the executive committee looks unbalanced: three relatively senior academics, all computer scientists without any background in engineering.²⁶ Though all had considerable experience in industry, that experience would not have been obvious to anyone looking at affiliations (and, in any case, was experience as “software engineers”, not engineers strictly so called). Since Rogerson was not an American, he gave the EC an international flavor; and since he was in information management, he might also seem to offer a management perspective (as indeed he did). But Miller was almost Gotterbarn’s twin: an American professor of computer science.

The first paragraph of the February 4 memo explains that what follows is “a copy of the code” (Version 1) with comments integrated. The source of each comment is indicated by initials: DF (Frailey), MS (Shaw), ED (Mechler), and so on. Most of the commentators are familiar from Chapter 5, but two are new: sr (Rogerson) and DS (“second pass comments” from Gotterbarn and Rogerson—“D” for “Don” and “S” for “Simon”). Some of the comments “raise difficulty” because “they are based on a miss-reading of the code” or are “just plain wrong” (for example, Shaw’s view of “bribery as good international business practice”). The task of the EC is “relatively quickly” to restructure the code “a. to improve it, b. to address objections raised, and c. [to] put together changes suggested with reasons for each and circulate to the task force.” Gotterbarn then sketched his own ideas for improving the code:

A.preamble introduction.

- i. revision of the introduction to include what we consider main functions of code
- ii. indication that code is not a list, black letter law, as in BCS code—put in preamble
- iii. make clear structure based on relationships²⁷
- iv. make clear that client & employer may not be synonymous, may even mean professor and student.
- v. include all people related to SE, like academics
- vi. YOUR ADDITIONS OR DELETIONS FROM THIS LIST

B. bigger issues

- i. need to clearly bring QUALITY into the code—maybe in places that mention efficiency, for example 1.13 replace "productivity" with "quality"
- ii. need to bring in teamwork and management issues
- iii. a professional owes higher order level of care
- iv. relate all stakeholders not merely "end users"
- v. address "intensity issue" by replacing "assure" with different terms
- vi. YOUR ADDITIONS DELETIONS FROM THIS LIST

C. minutia

- i. change "RULES" to "PRINCIPLES"
- ii. will we interleave examples 1.1,1.2 into list of principles, NSPE model or make the examples a separate list of guidelines ACM model.
- iii. YOUR additions or deletions from this list

Clearly, Gotterbarn has gone beyond resolving the issues that Frailey, Shaw, and members of the task force had raised the preceding autumn. Who, for example, had asked whether “client” might mean “professor” or “student”?²⁸

Even the two (substantive) “minutia” suggest that Gotterbarn was planning to rewrite the code in ways that went well beyond the criticism received in October and November. Both the “minutia” are important changes in architecture. The change from “rules” to “principles” is not merely the substitution of a three-syllable word for a one-syllable one. It also substitutes a more ambiguous term for a less ambiguous one. I had called the general standards “rules” to stress that they were as binding as the particular standards, merely more general forms of the same sort of standard; I had used “shall” in them rather than “should” for the same reason. For many people, “principle” suggests a standard of a different kind from “rule”, for example, a standard that should have weight in deliberation, not—like “rule”—one that, while open to interpretation, is binding, something one either obeys or violates.²⁹

While Gotterbarn seems unaware of this possible understanding of “principle”, he would soon (February 10) offer his own suggestion for distinguishing between “rule” and “principle”: “these [the Rules] are not really single rules but statements that embody a set of activities. This change [to Principles] lessens the possibility of misreading of the code as a checklist.” Since all rules embody a set of activities, Gotterbarn has not actually explained the distinction. He has simply offered the name “principle” as a better way than “rule” to lessen the likelihood that the standards in question will be used as a checklist. What reason he had to believe that the change of name would actually accomplish that end he did not say.³⁰

Similarly, segregating the particular standards (“examples”) in a separate section, as the ACM code does, might (as Gotterbarn also claims) lessen the likelihood that the Principles will be used as a check list. But it might not. Other outcomes seem equally probable. For example, a separate section for the principles might invite hasty users to forget about the particular standards altogether or suggest that the particular standards be treated as mere commentary rather than as binding rules, something to look at only if one has doubts about what a principle on the “check list” means.

That Gotterbarn would describe such architectural changes as “minutia” is surprising. The “minutia” in question might well make the difference between a code widely used and one soon forgotten. Gotterbarn’s later extensive reworking of the code, whatever its merits, seems to show that he did not regard these issues as “minutia”. Perhaps “minutia” is no more than an unfortunate “slip of the pen” (or whatever the word-processing equivalent is). That Gotterbarn would make empirical claims about how the code is likely to be used without evidence is not surprising (even if interesting). Though we know little about how codes of ethics are used, claims about the heuristic advantages of writing a code one way or another are common (and, in my experience, often wrong—or, at least, implausible upon examination).³¹

7.5 The great revision

The next few days would have been hard for Gotterbarn even without trying to straighten out Mechler’s table. Gotterbarn, apparently with much help from Rogerson, rewrote the entire code (drawing on responses received in October and November), annotated the changes, sent the annotated revision to Miller and Rogerson for comment, and revised accordingly. Since Miller was in Illinois, not working down the hall (as Rogerson was), we have some idea when

Gotterbarn completed Version 2. On the morning of February 10, Miller sent in his last comments (responses to Gotterbarn's replies to his earlier comments). Some reveal how important Rogerson was. For example, Miller had found several provisions with "outwith" in them and confessed to not knowing what the word meant. Gotterbarn had responded, "Simon's contribution—a scot[t]ish idiom, meaning 'standing outside of'. He thought it would add an international flavour to the code and determine if anyone read this far." Miller's February 10 reply is: "HA, HA... I passed the test! International flavor or international obscurity?"³² Writing a code of ethics, though serious work, need not be solemn.

Some of Miller's comments reveal the process of choosing words by which a code comes into being. For example, Miller had commented (on the always troublesome) 1.14 ("Avoid fads..."): "Are you going to include a citation to a handbook of 'standard' software engineering practices. I want that reference." Miller's request for a reference was, of course, ironic. The Software Engineering Institute was, in effect, working toward such a handbook piecemeal. Any handbook was (as Miller knew) decades away (and, like engineering handbooks, would never be complete or entirely up to date). The Body of Knowledge Task Force would, at best, offer only relatively uncontroversial standards as part of the "body of knowledge" all software engineers share (a basis for identifying those who know the "basics" rather than for guiding the experienced). Some practices might, though popular, be novel enough not to belong in the body of knowledge or even in SEI standards and yet not be what anyone would understand as a fad. Gotterbarn therefore answered Miller:

We have struggled with that one...We received major attacks on "fads".

Perhaps answer your "standards" question in terms of "what is publicly defensible" or in terms of results from the skills task force. I know neither of these is a good answer, but they are answers other technologies would use in defending that a practice was 'standard'.

Any ideas how to reframe this point?

Miller's February 10 suggestion is to rewrite 1.14 to read (something like): "software engineer should follow industry practice that, in his/her judgment, are most appropriate for the task at hand. Novel and experimental techniques may be appropriate, but not for the sake of novelty."

That evening, barely a month after he arrived in Leicester, Gotterbarn sent the task force (an annotated) Version 2 of the "Code of Ethics for Software Engineers". Having had enough trouble with attachments when he sent out the table of code references, Gotterbarn did without attachments, dividing his message into three emails (none large enough to clog a standard mailbox). The first was a page-long covering letter with "Principles 1-2"; the second, "Principles 3-4; and the third, "Principles 5-7".

The division into parts did not work quite as planned. Gotterbarn sent off his emails just before 7:00 PM Leicester time. By 6:00 PM (EST), that is, several hours later, Norman used the listserv to tell Gotterbarn he had all three (February 10, 1997).³³ Next morning, however, Milton Fulghum (Missouri) emailed (through the listserv): "I have received parts 1 and 2 but not three." About noon (February 11, 1997), Langford (Kent, two hours by train from Gotterbarn) sent a similar response to Norman (through the listserv): "O fortunate one—I managed to get (2) and (3), but am still anxiously waiting for (1)..." Late that afternoon, Gotterbarn advised

“patience...[the] net being what it is, the messages are taking interesting routes to get to you all.” He gave one example to make the point:

I mailed them [the three emails] from the Centre for Computing and Social Responsibility here in England. The director [Rogerson]—office across the hall from me—did not get part one until 2:00 pm this afternoon....[while] a research associate [Fairweather] upstairs had all three this morning (GMT).

Gotterbarn urged everyone to “tackle, grapple with, the parts you do have :-). I appreciate your enthusiasm, but we are actually slowing down their arrival filling the net with these messages (me too, I guess with this message).”

By the next morning, everyone who had any part of the tri-partite message had all three. But there was at least one problem of another kind, the mailing list itself. Weil emailed Gotterbarn (February 12, 1997): “I don’t understand why my colleague Michael Davis has not received the recent flood of e-mail. We thought csep@charlie.cns.iit.edu was on the [list].” Gotterbarn responded that “[Davis] has always been on the list as weil@charlie.acc.itt.edu [sic] and is still on that way.”³⁴ He recalled “we were having some trouble with his address last year and...per instructions we changed his address to yours, only at .acc.” Gotterbarn then promised to change the address if it was now not working. He went on to say that he “would appreciate his comments and yours on the current version of the code and our plans”³⁵ and concluded by asking “Are you still tracking us for NSF?”³⁶

Once complete, the tri-partite message was an impressive, even if incomplete, document. It was seven single-spaced pages long. Many of the code’s seven “Principles” or specific clauses had bracketed numbers inserted (1-19). Each number corresponded to a comment at the back of the email (with numbering beginning at 1 in each email). Each number either marked a change already made or identified a place where a change was “under construction”. The comments explained a change and sometimes provided a justification (though some of the Principles also had explanations preceding them). There was no “Introduction” because “changes to the Preamble are being worked on now.”

The covering letter (February 10, 1997) not only explained these matters and urged everyone to read through the draft “in the next few days” but to “give us your comments” and “your suggestions for improvements”. The letter also included two substantive paragraphs explaining Version 2’s architecture. (These paragraphs used the future tense, suggesting that they had been pulled from an earlier planning document without editing.) The first described “v2.0” as having an architecture “similar to version 1.0.” It “will start out” with a “preamble or introduction explaining the code and [providing] guidelines on its use.”³⁷ Following the preamble will be “a series of keyword principles describing a software engineer’s obligations in a variety of relationships.” (Apparently, what Gotterbarn meant by “keyword principle” is what most people would call a “title” or “keyword”.) Associated with each keyword is a “high level description of the software engineer’s obligations [what Gotterbarn elsewhere calls a ‘principle’].” Following each of these is “a series of clauses detailing some of the obligations under this Principle.” These obligations “are taken from all three levels of obligations in a professional code of ethics” (human, profession as such, and software specific). Gotterbarn had *not* reworked the Code to mirror the three levels.

The second paragraph calls this architecture of “major principles followed by explanations or examples...fairly common.” It then gives two carefully chosen examples to prove the point: “it is used by the NSPE and the British Computer Society”. The mention of the British Computer Society protects Version 2 from being treated as an engineering-style code rather than a computing-style code (or as an American-style code as opposed to an international code). While politic, the claim that Version 2.0 resembles other codes in this way is only very roughly true. Version 2.0 (and its predecessor) are alone (at least among the codes Gotterbarn discusses) in *explicitly* treating the particular rules as examples (special cases) of the general rule under which they are listed. In most codes having particular rules listed under a general one, the relationship is left to interpretation. Also “quite common”, according to Gotterbarn, is the use of the preamble to inform users that “the code should not be considered as a complete list of professional ethical obligations”, for example, “all codes in the architecture table follow this format”. In fact, that is not true of any of the engineering codes cited—though it is certainly reasonable to interpret them as having such a provision “implicit”.³⁸ Version 2.0 (and its predecessor, 1.0) had, in this respect, joined a relatively small number of recent codes (of which the ACM and BCS codes are two important examples).

Code 2.0 (and, presumably 1.0 as well) “differs from most codes” in “its use of keyword principles”. Most codes do title their major sections (though some, like the 1979 IEEE code) merely give “reference titles” such as “Article I”, “Article II”, and so on. Some, such as the Australian Computer Society Code of Ethics, also have “keyword” titles (such as “Priorities”, “Competence”, and so on) and general standards (such as “I will serve the interests of my clients and employers, my employees and students, and the community generally, as matters of no less priority than the interests of myself and my colleagues”) as well as particular rules under them (for example, “I will endeavour to preserve the continuity of computing services and information flow in my care”).³⁹ The “SE code differs [from other codes, according to Gotterbarn,] in that its keywords are tied to professional relationships”. This is important because “[works] on computer ethics have found this organization [by professional relations] useful in covering the topics of professionalism and useful as a mnemonic [mnemonic] device.”⁴⁰

The comments suggest that Version 2.0 is, overall, a less demanding (“more aspirational”) document than Version 1. In many respects, that is true.⁴¹ So, for example, comment 6 notes that “Strive to” has replaced “Assure that they” in 1.02 (“Strive to understand fully the specifications for software on which they work”) to “placate those who appeal to impossibility of precise specs”; and comment 10 notes that “Aspire to” has replaced “Assure that they” in 1.12 (“Aspire to identify, define and address ethical, economic, cultural, legal, and environmental issues”) as part of a general program to use a “more aspirational and intentional phrase in these level 1 type obligations in 1.10-1.14.”⁴² Again, 2.02 (“Approve software only if they have a well-founded belief that it is safe, meets specifications, and has passed appropriate tests....”) is the result of three amendments: One substituted “well-founded” for “well-documented”. The second deleted the word “all” before “appropriate tests”. The third substituted “appropriate” for “proper”. Note 15 points out the first change but does not explain it. For software engineers bedeviled by lack of documentation, the substitution of “well-founded” reduces Version 1’s overall emphasis on documentation. A programmer need only think she has good reason to approve; she need not document her reasons. When a question arises a year later and she is gone to another job (or forgotten what she thought at the time), how will anyone know what her belief was—what she considered “safe”, what she thought the specifications were, or

what tests (if any) she thought “appropriate”? Similarly, note 15a explains that “all” was deleted “to avoid the suggestion that exhaustive testing is required” (just what Mechler thought had been avoided by having “as appropriate” precede all clauses under Rule 1). But, in avoiding the “suggestion” of exhaustive testing (a standard generally too high to be practical), the deletion of ‘all’ had the effect of permitting a software engineer to do no more than *some* appropriate testing, something short of all the tests that should (together) be done.⁴³

Though the overall tendency of the proposed revisions was to make the code less demanding (more “aspirational”), some changes seem to add to its demands. For example, Rule 2.05 (“Be fair and truthful in all public statements”) had become “Be fair and truthful in all statements, particularly public ones, concerning software or related documents.” Comment 18 explains that, as originally written, 2.05 “indicated that it was ok to lie in non-public statements” (on the questionable but not unknown interpretive principle that whatever is not expressly forbidden is allowed).⁴⁴ Comment 18 apparently overlooks what the code’s introduction says about the status of particular rules. They are instances of the general rule. So, for example, 2.05 is an instance of “2. PUBLIC Software engineers shall, in their professional role, act only in ways consistent with the public safety, health and welfare.” Any non-public statement inconsistent with the public safety, health, or welfare is also disallowed. What the revised 2.05 seems to disallow (that the original did not) are non-public lies concerning software or related documents that are consistent with the public safety, health, and welfare. For example, 2.05 now seems to disallow lying in non-public statements to tyrants or unscrupulous employers even when such lies serve the public safety, health, and welfare. A software engineer must be “truthful” even with them. Since Principle 2’s “In particular” has no “as appropriate”, Version 2.0 therefore sets a higher standard than Version 1.

Some changes that seem intended to raise standards may in fact lower them. Consider, for example, comment 8 under Principle 3 (Judgment). The comment notes the need for some “social clause” in 3.06 (“Participate in no decision of a governmental or professional body, as a member or advisor, concerned with software, or related documents, in which they, their employer, or their client has a financial interest”). “[As] this clause stands judgment is merely a technical thing and does not involve social issues.” The comment then offers the following as possible substitutes for 3.06 (apparently intending to raise the standard):

Only make judgments which can be publicly justified as being in the best interest of quality of life, and the environment.

All technical judgments should be tempered and guided by the need to support and maintain human values.

Be aware that all technical judgments impact other human beings. They have stakeholders other than employers, clients, and users.

Gotterbarn seems to have understood that 3.06 (and, indeed, all of Principle 3) is concerned with maintaining independent professional judgment (that is, avoiding conflict of interest), not with substantive issues discussed elsewhere in the code.⁴⁵ What he does not seem to have understood is why it is important to have a separate section on technical judgment after two sections (“Product” and “Public”) concerned with the very substantive issues he proposes to incorporate

into 3.06. The avoiding of adverse financial interests is a *means* of helping to ensure that the judgments in question “can be publicly justified as in the best interests of quality of life” and are “guided by the need to support human values”. Substituting the general imperative for a particular means, allows the end to be pursued by other means instead (for example, by giving to the poor some of the profits derived from keeping the adverse financial interests). The point of 3.06 is that there is no substitute for avoiding this sort of conflict of interest. Good intentions (and even compensatory good deeds) are not good enough. Gotterbarn’s recourse to good intentions as a substitute for the flat prohibition of certain conflicts of interest is thus a weakening of the code even though intended to strengthen it.

Some changes are hard to categorize on the dimension of more-or-less demanding because their overall effect is hard to gauge. For example, Version 2.0’s Principle 1 has “free of error” where Version 1 had “free of significant error.” The comment on this change explains that “[we] felt uncomfortable with the phrase...It might be used as an escape clause—‘I had lots of errors but didn’t think any of them significant’.” “We” tried to compensate for this strengthening of the code by specific changes in Principle 2 (“PUBLIC Software engineers shall, in their professional role, act only in ways consistent with the public safety, health and welfare.”) But there “is still some concern with this change [deleting “significant”] because it is not possible to write error free code. Any suggestions for balancing these concerns?” The overall effect of this combination of changes may or may not be a change in what is actually demanded. What seems clear is that Gotterbarn, Rogerson, and Miller (“we”) are aiming at something very close to the balance that “free of *significant* error” was intended to achieve. (Eventually, they would seek to achieve that effect by radically different language.)

Other changes are hard to assess on the more-or-less demanding dimension for other reasons. For example, the third change in 2.02 substitutes “appropriate” for “proper”. The explanation for this change (here and elsewhere in the code) is that “appropriate” is “less vague” than “proper”. There are “technical answers to the question ‘which methods are appropriate?’”—but, apparently, not to the question, “Which methods are proper?” Many readers may find the claim that “appropriate” is less vague than “proper” novel and suppose that all that has happened is that a “\$3 word” has replaced a shorter one. Others may suppose that technical standards set what is “proper” but interpretation (professional judgment) must decide what is appropriate. Given that supposition, the change from “proper” to “appropriate” would be a weakening of the code (a substitution of individual judgment for common standard) rather than (as the comments claim) a strengthening. Some word must be chosen, and all will be open to misinterpretation. The most precise word is, all else equal, the best. But what makes a word the most precise? And how are the drafters of any code to know?⁴⁶

Another example of hard-to-assess changes is the general replacement (in Principle 1) of “assure” with “ensure” (where “assure” is not replaced by “strive” or some other less demanding term). But here the claim to know which word is better relies on a linguistic authority (the *Oxford English Dictionary*). Comment 3 (under Principle 1) defends the replacement as a weakening of the obligation: “‘ensure’ is less legal than ‘assure’ which carries a formal guarantee and legal connotation (according to the OED); whilst ‘ensure’ means to strive to make things happen and no formal guarantee is implied.”⁴⁷ Appeal to the OED may seem parochial. An American who looks up “ensure” in Webster’s (or one of its American competitors) will find the entry under “ensure” to be: “same as *insure*” (a formal legal guarantee); or “make sure or certain” (something distinctly stronger than “strive to make happen”).⁴⁸ So, in the US at least,

replacing “assure” with “ensure” may not weaken the obligation (or even avoid the hint of formal legal guarantee). If anything, the replacement strengthens the language, substituting the robustly objective “ensure” for the faintly psychological “assure”. And, in fact, I had chosen “assure” over “ensure” because I did not want to suggest “insure”. I too was concerned to avoid the suggestion of “formal legal guarantee”. What I had overlooked is that code-writers almost universally prefer “ensure” (even if they do not know why). Here, as in many other places, there is agreement about the end in view in the choice of a certain word but disagreement about the meaning (or likely impact) of the word in question. More, interesting, though, is that OED entries for “ensure” and “assure” are similar to Webster’s.⁴⁹ The OED does not in fact appear to explain the preference for “ensure”. Did anyone on the EC actually check the OED?

Amid all these changes in wording was one big structural change. The third part of the three-part email proposed to add a whole new section, “Principle N Management” (“N” being a placeholder until the new section received a number determining its place in the code). The principle itself was only partly stated: “A software engineer in a management or leadership capacity shall....” But under this half-stated principle are nine clauses (N6.06-N6.15). Most, such as N6.06 (“Assure that employees are informed of standards before being held to them”), are simply pulled from the clauses under Version 1’s Principle 6 (“Colleagues”). Some are slightly revised. For example, N6.13 differs from Version 1’s 6.13 only in having “remuneration” in place of “compensation” (“Offer only fair and just remuneration”). (The explanation for this change is that “in Europe [‘compensation’] means ‘an item given to redress a wrong done you’”).⁵⁰ One clause is entirely new (“N6.15 Not ask an employee to do anything inconsistent with this code”). Preceding the particular clauses is a warning that they are “under construction”. Clauses yet to be formulated should include “enable, encourage, follow good practice”, “provide direction to good software engineering practice”, and something about “Team building, Communications, and Good practice”. These clauses never entered any version of the Code..

The idea for a distinct section on supervisors, managers, or leaders may have come to Gotterbarn from one (or more) of five sources. First, Gotterbarn understood the code to be organized according to “relationships”. Management is an important relationship, one software engineers regularly find problematic. Insofar as Gotterbarn’s understanding of the code’s architecture is the source of Principle N, Principle N is evidence that theory can make an important contribution to writing a code (by calling attention to an option that we might otherwise overlook). Second, Gotterbarn may have gotten the idea for Principle N from the ACM Code. Unlike most other professional codes, it has a section on “Organizational Leadership Imperatives”. While Gotterbarn may have gotten the general idea from the ACM Code, he does not seem to have drawn anything more specific idea from it. None of Principle N’s clauses seem to share language with any of the imperatives in the ACM code’s (large) leadership section. Third, another possible source of Principle N is Rogerson. He had a special interest in the ethics of technical management. Fourth, the list of rules under Principle 6 had become significantly longer than any other. It therefore invited the question: can we divide this in some way? Fifth, and most probable, Gotterbarn may have gotten the idea from Milton Fulghum, a SEEP volunteer. Fulghum suggested dividing Rule 6 in just this way in his comments of October 22, 1996. Among the computer files Gotterbarn retrieved from George Washington University after his arrival at DMU was Fulghum’s email.⁵¹

7.6 Why the new Gotterbarn?

Just two days before the deadline Cabrera and Frailey had set Gotterbarn to deliver “a *complete* draft of the code of ethics and professional conduct”, Gotterbarn was sending off for comment a draft much rougher than the one he had started with—a code lacking a preamble, differing in many important ways from its predecessor, with some provisions still “under construction, and the whole untested by outside comment”—, a code no more “[rooted] in the codes currently in use” than its predecessor but now salted with question marks.⁵² Once a distant administrator, punctuating long silences with a new plan others were to follow and then waiting quietly for them to follow it, Gotterbarn had suddenly become active, inventive, and daring. He was doing much of the work himself. He was not simply revising the code to meet objections but rewriting the code as he thought it should be written. He was betting that he understood the politics of the Steering Committee, and of the two organizations that had joined to create it, well enough to undertake fundamental revisions at what must have seemed rather late in the game.⁵³ What explains this change in the way Gotterbarn worked? We may identify at least three (more or less) related possible explanations (in addition to having the more flexible schedule of a sabbatical leave).

First, Gotterbarn seems to have begun his co-chairing of SEEPP somewhat in awe of the IEEE-CS. At his first meeting with Barbacci and Melford, he had accepted the IEEE’s standard-setting procedure, the organization by task force and working group, and even a list of working groups having little relationship to his previous experience helping the ACM write its code of ethics. That, in the mid-1990s, a computer scientist would, all else equal, defer to engineers on how to organize a large-scale project is understandable. The reason for coining the term “software *engineer*” almost three decades earlier was precisely to point up the superior organizing ability of engineers, to invite programmers, computer scientists, and the like to learn from engineers how to organize a large project, how to complete it on time, within budget, and as specified. In addition, the joint project had begun as an exclusively IEEE-CS undertaking. ACM had joined after the Steering Committee had begun work. Less clear about what it wanted the project to accomplish, the ACM had not demanded that its procedures be followed. Gotterbarn had begun work in what must have seemed an engineering environment. He needed time to learn his way around, to see the limits of the IEEE method, and to gain the confidence of the IEEE-CS appointees on the Steering Committee.⁵⁴

Second, once organized, SEEPP could only work (Gotterbarn supposed) if he and Melford agreed. Gotterbarn could not act on his own (except as leader of his working group). Melford could prevent Gotterbarn from acting simply by falling behind in answering his email. He had a (unintended) “silent veto”. While Gotterbarn accepted the IEEE’s standard-setting procedures, including the division of labor among working groups, and Melford had time to meet, the requirement that the co-chairs agree did not matter much. SEEPP was able to produce the documents that the IEEE procedures called for—at something like the rate typical of the IEEE standard-setting process. However, once Melford’s business began to overwhelm him, leaving him less and less time for outside activities, his “silent veto” became important. However much Gotterbarn doubted the value of the complex structure to which he had initially consented, it was now something he could not change unilaterally. He could not even (he supposed) properly email SEEPP, much less call a meeting, without Melford’s express consent. He was like a fish in a frozen lake. When Melford resigned as co-chair, the lake thawed.

Third, Gotterbarn could work on his own through his own group, Professional Competence, but he could work there only within the group's limited mandate. His working group could not write a code of ethics (or a code of professional practice), only a part of it ("avoid generalization of expertise, keep current, keep staff current, truth about skill, what should be produced, appropriate knowledge base").⁵⁵ What Gotterbarn did ask, which was all that he could ask, did not much interest many of the volunteers. They slowly drifted away. Though he could see what was happening, he could (it seemed) do nothing about it. As SEEPP's co-chair, he had to be careful about organizational etiquette. If he ignored Melford, or just offended him, he might alienate IEEE-CS, risking the entire undertaking. Mechler succeeded where Gotterbarn failed precisely because he ignored the organization in a way Gotterbarn could not. Or, rather, Gotterbarn could succeed within the structure he had accepted only by using a device familiar to software developers, the "skunkworks", that is, "a group of people who, in order to achieve unusual results, work on a project in a way that is outside the usual rules."⁵⁶ Had Mechler not rushed in on his own, Gotterbarn would have had to invent him. Once Melford resigned, there was no need for a skunkworks. The Steering Committee, having learned to trust Gotterbarn, gave him the power to create a structure within which he could work. He then worked more or less as Mechler had, though within the rules and with a speed even Mechler never exhibited.

7. APPENDIX: Comparison of Version 1.0 with Other Codes

From: IN%o"PRFCMP-L@UTKVM1.UTK.EDU" "Professional Competence Standards Task Force" 3-FEB-1997 07:10:33.22 To: IN%"PRFCMP-L@UTKVM1.UTK.EDU" "Multiple recipients of list PRFCMP-L" CC: Subj: Text version of table

Return-path: <@uga.cc.uga.edu:owner-prfcmp-1@UTKVM1.UTK.EDU> Received: from uga.cc.uga.edu by minna.cns.iit.edu (PMDF V5.1-4 #19448) with SMTP id <O11EYFUVB5C8ZK3P4@minna.cns.iit.edu> for CSBURNSTEIN; Mon,

Sorry about the problems with the early version of the table. Only one person (with a MAC) was able to read it. To avoid all difficulties with versions of word processors etc. I spent a pleasant ground hog day (a special holiday in the colonies) rebuilding the table.

Explanation of the table: The first column refers to sections of the SECV1. Only those items which closely matched the statements in SECV1 are referenced. If the referenced item in the other code is almost a direct quotation, then the reference appears in single quotes.

The codes used for comparison, listed in alphabetical order, are: The American association of Engineering Societies, Model Guide for Professional Conduct(AAES); Accreditation Board for Engineering Technology's, Code of Ethics for Engineers (ABET C of E) and Guidelines for The Fundamental Cannon of Ethics (ABET G); The Association of Computing Machinery's Code of Ethics(ACM), and Guidelines for Professional Conduct(ACM G), The British Computer Society Code of Conduct (BCS C of C); The British Computer Society , Code of Practice(BCS C of P): The Institute for the Certification of Computing Professionals (ICCP); The Engineer's Council for Professional Development, Faith of the Engineer(ECPD Faith); The Institute of Electrical and Electronics Engineers, Code of Ethics(IEEE C of E); The National Society of Professional Engineers, Code of Ethics for Engineers (NSPE C of E), and the Project Management Institute "Code of Ethics for the Project Management Profession" (PMI). Several of these codes did not contain section and paragraph numbers, so the following reference procedure was followed. If the document was not divided into sections, its paragraphs were simply numbered sequentially starting with 1. If the code was divided into sections and paragraphs. The paragraphs were given an alphabetical designation and the paragraphs within each section were numbered sequentially starting with 1. Most of the codes are in Ethics in Engineering, Martin and Schinzinger, and the other codes are on the NET Reference Table Comparing Software Engineering Code v 1.0 to Other Codes of Ethics. version 1.0 January 1997 Software Eng. C of Ev1:

	AAES	ABET	ABET-G	ACM	ACM-G	BCS-C	BCS-P	ICCP	ECPD	IEEE	NSPE	PMI
Product												
1.0 1				3.4								II.b
1.02	A			3.4								II.b
1.03	C1	B.2 2				20	1.3	2.5		'6'	'II.2.a'	I.b
1.04			1.c.1	3.4								
1.05				1.2					2			II.b
1.06									2			II.b
1.07							2.4		'3'			II.b
1.08		1.c.1		3.4							'II.2.b'	II.b

1.09		1.c.1	2.5									II.b
1.10				1.8, 3.5 1.7	1.8 3.5	2	3.5					I.b
1.11					1.7	2	4.6					
1.12	C2, C4	A.1	1.a, 1.c.1 4.j		2.3 3.1	1.1, 1.4	1.7, 8			8		
1.13									2			
1.14				1.2								

	AAES	ABET	ABET	ACM	ACM	BCS	BCS	ICCP	ECPD	IEEE	NSPE	PMI
Public	C.3 C.9	A1, A2	1.b, 1.c.3 3.b	1.3	1.1 1.2 2.5	7		2.7 3.7	3 3	1 '1'	II.1.a	IV.a
2.01			1.b	1.2		1				1	II.1a	IV.a
2.02			1b, 2.c		3.2	21	1.3	2.7			II. 1.b'	I.b
2.03					1.2					7		
2.04	C.3,	B.3	1.b, 2.c, 3.b		2.4 3	17		3.4		3	I.3, 'II.3.a'	
2.05	A,C9	B1, B3	1, 1.b						3		II.1.a	
2.06	C.8				1.2, 2.4, 2.6	21			3	1	I.1	I.a
2.07												

	AAES	ABET	ABET	ACM	ACM	BCS	BCS	ICCP	ECPD	IEEE	NSPE	PMI
Judgement									2.1			
3.01		A.2,	2.b	2.5	2.5						II.3.c	
3.02		A2, B4		2.6		7				'4'	II.5.b	
3.03	C.5	A.2,	4.c, 4.e	2.6		7					4 II.4.c, II.5.b	III.d
3.04	C.5	A2, B4	4.c	2.6		7	4				II.4.b	
3.05		B.4	4.a, 4.d, 4.g	2.6		8	4				'II.4.e'	
3.06		B.4	4.f	2.6		12	4				II.4.d	
3.07	C6, C9	B.4	1.c 3.d, 4.a	1.3 2.5		5.7, 9,22			2	'2'	II.4.a	III.a

	AAES	ABET	ABET	ACM	ACM	BCS	BCS	ICCP	ECPD	IEEE	NSPE	PMI
CIntlEmp	A.2 C.1	B2, B5	2, 3.c, 5.b	2.2	2.2		1.3	2.5		6	'II.4' I.2	I.b
4.01												
4.02				1.5							II.1.b, II.2.c III.7.c	III.a
4.03			4.I.3 5.o	3.3	1.5, 1.7	2						III.a
4.04				1.5	1.5							
4.05	C.5	B4	4.I, 4.i.	1.9, 11.8	1.9	2	4.5	2.1			II.1.c, III.4	

4.06				2.5								
4.07		4.h		1.5	2.5	26					'III.1.b'	III.a
4.08								3.6			I.4	III.a
4.09					2.6						'III.1.c'	III.a
											I.4	III.a

Profession	AAES	ABET	ABET 4.I.4, 4.m 4.b	ACM	ACM	BCS	BCS	ICCP	ECPD 1	IEEE	NSPE	PMI
5.01											II 1.d	
5.02		A4, B6	6	4.1			13			10		I.e
5.03	C7, C10		6.a	4.1			13		3	10		I.e
5.04	B, C.9,	B6		3.1								I.e
5.05		A4, B6		4.2	1.2, 2.5		13		3		I.5, II.1.e	
5.06			1.f, 6		1.2					1	II.1.a	I.a
5.07			1.c, 1.d								III.6.b	
5.08	C.3		1.c.	22.5		10		3.4	3		I.5, III.3.a	
5.09		B.6	5.b						1		III.1.f	
5.10	A,B, C9, C10			2.3	2.3			3.2				I.b
5.11			4	3.6							'III.2.a'	
5.12					2.7	14		2.2			III.2.c	IV.b
5.13 C.7			1			13	1.2	3.1	5		III.11a	
5.14			3.a, 7.f				8,15			4	III. 11.b II.5.b	

Colleague	AAES	ABET	ABET 7.c	ACM	ACM	BCS	BCS	ICCP	ECPD	IEEE 8	NSPE	PMI
6.01	C.7	B.7	5.a	3.6	3.6	18	1.2		4	'10'	III.2	II.g
6.02		B.7									'III.8.a'	
6.03	C.8	B.7	7.a	1.6	1.6	16				7	III.10.a	II.f
6.04			4.o	16,19	2.6					7		
6.05	C.8		5.i, 7.e						3			II.c
6.06			3.c, 5.m		4.1		2.1	2.7			III.11.e	II.c
6.07							2.1, 4.5	2.7			III.11.e	
6.08							1.3 2.2				III.11.d	
6.09									5			II.c
6.10											III.10.c	
6.11					16						III.11.e	
6.12			4.k	2.7							III.11.e	
6.13			5.d								III.6.b	
6.14			4.n, 7.i		16					8,9		

AAES	ABET	ABET	ACM	ACM	BCS	BCS	ICCP	ECPD	IEEE	NSPE	PMI
------	------	------	-----	-----	-----	-----	------	------	------	------	-----

Self			7.g				1.1			6		
7.01	B, C7	A3, B7		2.	2	18	1.1	3.1	4	6	III.11a	I.c
7.02	B, C7	A3, B7		2.2			1.1	3.1	4	5	III.11a	I.c
7.03	B, C7	A3, B7	7	2.2			1.1	3.1	4		III.11.a	I.c
7.04	B, C7	A3, B7		2.2			1.1	3.1	4	5.6	III11.a	I.c
7.05	B, C7	A3, B7		2.3		3	1.1	3.1			III.11	I.c
7.06		A3, B7		4.1	4.1					10		I.c

*****TEMPORARY ADDRESS CHANGE***** Don Gotterbarn Centre for Computing and Social Responsibility DeMontfort University, School of Computing Sciences Leicester LE 1 9BH, UK

From: IN% "PRFCMP-L@UTKVM1.UTK.EDU" "Professional Competence Standards Task Fo r with spool id 5129 for PRFCMP-L@UTKVM1.UTK.EDU; Mon, 03 Feb 1997 09:17:10 +0000 Date: Mon, 03 Feb 1997 14:15:14 +0000 From: Donald Gotterbarn <dgot@DMU.AC.UK> Subject: Re: Code Annotation In-reply-to: <c=US%a=%p=EQT%1=ERIXCHANGE-9701301453512-8 Ed, As regards the schedule, I did draft a detailed schedule for completion of the project as was requested by the Steering committee. Until that schedule's details and the requested support are approved by them, I did not feel it was right to distribute it widely, but instead listed in general terms those steps which I hope will lead to a successful completion of our tasks.

I understand your concern and appreciate your enthusiasm, but I did not want to distribute anything which was not agreed upon.
regards, don

NOTES

¹ Gotterbarn Email (February 12, 2004) Attachment.

² Gotterbarn's computer arrangements were more complicated than this description suggest: "I had my laptop—which was a faster machine than the one supplied by DMU, so I brought it every day and did my work on it. I would then either transfer the data to the DMU machine and send it. Or I would go home in the evening and use my modem to connect and send messages." Gotterbarn Email (February 12, 2004), Attachment. Hence, Gotterbarn could send and receive email from home in the evening or on weekends.

³ SEEPP's working groups had, of course, long since ceased to matter; anyone who did notice the reorganization was unlikely to care. Gotterbarn therefore had no reason to emphasize the reorganization. Compare Gotterbarn: "I formed a single committee consisting of the Prof Competence people and the working groups leaders from the other groups, hoping they would bring their functioning groups members along with them. PRF-Comp was just a mailing list, a tool that was there and that was useful. It existed so it was easy to add people to the mailing list. There were no hidden motives!! You make something of nothing. I did a complete reorganization. A Single purpose group for single –clear task assigned by steering committee. If I had a functioning mailing list labled 'happy day' I would have used it." Gotterbarn Chapter6cmt (October 31, 2004).

⁴ See email from Gotterbarn to Mechler (February 3, 1997): "I did draft a detailed schedule for completion of the project as was requested by the Steering Committee. Until that schedule's details and the requested support are approved by them, I did not feel it was right to distribute it widely, but instead listed in general terms those steps which I hope will lead to successful completion of our task." This seems to me to be a change in the way Gotterbarn (and Gotterbarn-Melford) had operated until 1997. Gotterbarn-Melford had often announced schedules without Steering Committee approval. What explains the change? "By getting approval for a plan," Gotterbarn suggests, "I could proceed full speed ahead without asking permission at every turn." Comments on Chapter 6 (September 23, 2003). Why did Gotterbarn think he would otherwise need "permission at every turn"? The Steering Committee has already authorized him "to drive the task force to completion". The Steering Committee had also said he would distribute the plan "next week" but not that he needed anyone's permission to carry it out: "Other activities for the remainder of the year will be outlined in a plan that Don will distribute next week."

⁵ The two references to Gotterbarn's own work are his entry, "Software Engineering Ethics", in the *Encyclopedia of Software Engineering* and one of the papers he gave at PASE'96, "Software Engineering: the new professionalism".

⁶ Actually, Gotterbarn gives the publication date of Johnson's anthology as "1996" rather than "1990" (a typo or perhaps a misreading of a handwritten "0" as a "6"). The first edition of the Martin and Schinzinger text came out in 1983; the second edition (1989) was basically the same book. Both references were, then, substantially less current than the stated publication dates suggest.

⁷ Comments on Chapter 6 (September 23, 2003). Gotterbarn\Gotterbarn memories of 1997 (9\1\2003).

⁸ “I was not writing for scholarly consumption. For details on this [his scholarly views], see my ‘The Moral Responsibility of Software Developers: Three Levels of Professional Software Engineering,’ *The Journal of Information Ethics* 4 (Spring 1995): 54-64.” Comment on Chapter 7 (September 21, 2005).

⁹ Charles E. Harris, Michael S. Pritchard, and Michael J. Rabins, *Engineering Ethics: Concepts and Cases* (Wadsworth: Belmont, 1995), p. 38 (a code of ethics as the solution to a coordination problem, something more than just a guide to individuals).

¹⁰ Was Gotterbarn aware that Mechler might object to the omission of standard-raising? Probably not, though he certainly had another opportunity to become aware of it. Four hours after sending off the statement, Gotterbarn got his George Washington email account to forward to his DeMontford address Mechler’s response to Shaw. He did this, it seems, as part of a larger effort to collect all comments on the code before preparing his revision. That same evening he retrieved the comments of Frailey, Fulghum, Langford, and Prinzivalli in the same way. (Two days later, he also retrieved Shaw’s original message.)

¹¹ There may have been an additional delay. At this time, my name had (again) dropped from Gotterbarn’s list. Any email I received from Gotterbarn was a copy Weil passed on—after checking to see if I had received it independently.

¹² In his comments on this chapter (September 23, 2003), Gotterbarn raises a question of professional ethics: “Don’t you wonder about the moral responsibilities of MD here?...Certainly he had responsibility as a professional to address that [that is, to correct what I considered Gotterbarn’s omission of an important purpose of codes].” The question is worth answering here because it underscores the difference between the (1980s) theories of professional ethics Gotterbarn took for granted and the theory I developed in opposition to it (and which, apparently, he was not aware of even in 2003). My position is that I have no obligations “as a professional [as such]”, only as a member of a particular profession (or as a moral agent, parent, promisor, or other particular relationship). What is my profession? I’m an academic, a member of the professoriate. That’s my only profession; philosophy is the discipline I practice, but not my profession (a special way to earn a living). To decide what my obligations as an academic are, I start with the code of ethics of the American Association of University Professors. I don’t find any requirement there that an academic must correct every error anyone makes with respect to his discipline, not even that an academic should correct every error a colleague (or student) makes. There may, of course, be obligations the code overlooks (the unwritten ones that derive from the written or have some other origin). But, any claim for one of those requires an argument—and, in this case, I believe there is none. What Gotterbarn seems to have in mind is a general obligation to be a Good Samaritan. I think there is no such general professional obligation. For more on this, see my *Ethics and the University* (Routledge: London, 1999), Chapter 4 (“Science: after such knowledge, what responsibility?”).

¹³ In 1997, “:-)” was still all that printed, though the meaning was a smile: ☺

¹⁴ Comments on Chapter 6, September 23, 2003. Gotterbarn later added: “One of the reasons for the delay [in recognizing that the problem was in the kind of type the printer used] is no one of us considered that an ascii table would be printed using anything but a fixed font. We did not look for the problem here because we thought people knew better. When it was thought that this might be the issue, it was stated in a way not to embarrass anyone.” Gotterbarn Chapter6cmt, October 31, 2004.

¹⁵ For a decade or so (until the mid-1960s), the ECPD code had the endorsement of most American engineering societies. The NSPE’s code is its direct descendent (and the ABET code, a radical reworking of it).

¹⁶ Schinzinger was a professor of engineering already close to retirement when he teamed with Martin, a young philosopher, to do the first edition of their popular text. He might well have remembered the Faith from his youth and suggested its inclusion.

¹⁷ February 3, 1997: “Most of the codes are in Ethics in Engineering, Martin and Schinzinger, and [sources for] the others are on the NET Reference Table Comparing Software Engineering Code v 1.0 to other Codes of Ethics, version 1.0 January 1997 Software Eng. C of E v1”. “Most” is both a (gross) understatement and a (slight) overstatement. The six codes from Martin and Schinzinger are *all* the engineering codes but only half of the total (counting codes as the table does).

¹⁸ Getting a copy of the code of ethics (“Ethics Policy”) for British engineering, for example, for the Institute of Electrical Engineers, was then, and remains, difficult. Only someone teaching a course in engineering ethics in the UK would be likely to have one easily available. Codes of ethics for computing are much more accessible in the UK. My concern in noting what Gotterbarn did not do is not to say he should have done more (under the conditions), but to be clear about what he did do (and therefore about what he did not do), facts relevant to assessing the changes he made.

¹⁹ Gotterbarn’s Comments on Chapter 6 (September 23, 2003): “I guess we didn’t do all those other things because we were paying attention to the TASK assigned- show that the draft code has roots in engineering and computing. Not law, medicine etc. The tasks as assigned were difficult enough to get done in the time frame without adding to the task.” Exactly.

²⁰ This use of “committee” may be a mere “slip of the pen”, but it may also be the first reference to the executive committee soon to be formally announced in the “Proposed Schedule”. See Chapter 8.

²¹ This, of course, is the only omission from Mechler’s list of eight (originally sent December 19, 1996, but resent January 30, 1997). I am, of course, assuming here (for the reasons given in 6.9) that the December email never reached Gotterbarn.

²² The question here is *not* who else contributed suggestions for revising the code. There were many (Frailey, Fulghum, Kanko's students, and so on, but not Mechler). Instead, the question is who provided comments or annotations concerning the "roots" of Version 1 in other codes.

²³ "Thanks for the credit—but Simon and Keith were always there." Gotterbarn Chapter6cmt, October 30, 2004. Yes, they were "there", Rogerson in person (when not out of town) and Miller by email. So, they certainly might have helped in constructing the table (hence, my hedging "more or less"). But no evidence survives of their participation in construction of the table—and neither of them actually recalls helping with it—though there is plenty of evidence of their helping on almost everything else. So, it seems to me, Gotterbarn really does deserve the credit I have given him ("more or less").

²⁴ Gotterbarn objects to this interpretation: "This is wrong—I do remember being greatly pleased by the work he provided. That is why I mentioned him specifically! GIVE HIM CREDIT! He was still working on the Code. YOUR THEORY MEANS I MUST HAVE LIED GIVING HIM SPECIAL THANKS." Gotterbarn Chapter6cmt (October 31, 2004). While I do not mean to suggest that Gotterbarn is here telling even a white lie, I do think his words give a misleading impression of what happened (however unintended). That Gotterbarn now remembers being please what Mechler's contribution is consistent with its merely providing a check on what Gotterbarn had already done (and with Mechler deserving the praise Gotterbarn gave him at the time). The timing seems entirely against Gotterbarn's claim that (in effect) five columns of the table are Mechler's work, not his.

²⁵ Gotterbarn\Version 1\Cmntster.

²⁶ Gotterbarn objects to my characterization of the executive committee: "It was a software engineering code—Miller an academic—does contracts for NASA etc., lots of experience on practical and academic side: same for Rogerson and Gotterbarn." Gotterbarn Chapter6cmt (October 31, 2004). True enough, but in addition to not being obvious to anyone who just looks at their institutional affiliations, these qualifications would, even if obvious, not satisfy an engineer that the perspective of engineers would be understood. Gotterbarn's criticism of this passage assumes that "software engineers", even if fully qualified to work on software, are necessarily engineers, an assumption most engineers would then (and even now) reject.

²⁷ Gotterbarn here misstates Version 1's structure. Five of Version 1's Rules (Public, Client and Employer, Profession, Colleagues, and perhaps Self) are "relationships" (in some sense) but the other two (Product and Judgment) certainly are not (or at least not in the same sense). Indeed, the long section on "product" is something unusual in codes. I had created it simply to hold a lot of suggestions not fitting any other category. I constructed Version 1 "from the bottom up" (ignoring the relational or function structures of earlier codes); Gotterbarn was constructing Version 2 "from the top down" (or, at least, wanted the Steering Committee to think of the code in that way). Seeing what I had done, he assumed I had worked the same way. In fact, my intentions (or method) could not be read from the results achieve (the code itself).

²⁸ Gotterbarn's comments on the first draft of this chapter (September 23, 2003) respond to my question "Who but an academic would ask that?"—"An academic who had actually read various international Computing codes of ethics—like the Australian Computer Society's code which specifically mentions students." The academic who actually read the code in question is not a hypothetic academic, presumably, but Gotterbarn (and perhaps the other members of the EC). So, it is worth pointing out that, when I checked the ACS's code at <http://www.acs.org.au/national/pospaper/acs131.htm> (September 23, 2003), I found a code that mentioned "students" (here and there) but never "professors". Its section 4.3 (Values and Ideals) is typical: "I must act with professional responsibility and integrity in my dealings with the community and clients, employers, employees and students." The idea of specifically mentioning "professors" (as well as "students") may have arisen from Gotterbarn's wide reading in codes of computing ethics and his wide experience with computer ethics more generally. What it did not arise from is any of the criticism he had so far received from anyone (or, at least, any that have survived on paper), even from criticism coming from professors and students—or from the code he actually cited.

²⁹ See, for example, Ronald Dworkin, *Taking Rights Seriously* (Harvard University Press: Cambridge, Massachusetts, 1977), esp. pp. 14-80; or the Walt Disney movie, *Pirates of the Caribbean* (2003), in which pirates unwilling to follow "The Pirate Code" when they should, say (something like), "Well, it's not so much a rule as a guideline."

³⁰ Gotterbarn's objection to "checklists" presupposes an answer to the prior question: *what is wrong with using a code of ethics as a checklist?* Using the code in that way certainly seems better than not using it at all (because, say, one does not know where to start and does not want to take the time to find out). Using the code as a (complete) checklist is a somewhat different matter. It too is better than nothing but definitely not as good as asking, after going through the checklist, whether there might be something the checklist should include but does not—or something that one should consider whether a code could include it or not. No doubt, there is something wrong with a question such as this: "Sexual harassment is prohibited, but what sort of harassment is not?"

³¹ For a critique of some of these claims, with references, see my "Three Myths about Codes of Engineering Ethics", *IEEE Technology and Society Magazine* 20 (Fall 2001): 8-14 & 22, reprinted (with some revisions) in *Profession, Code, and Ethics* (Ashgate: Aldershot, England, 2002), pp. 121-132.

³² Gotterbarn's memory of these events differs a bit from what my reconstruction: "When working hard many of us—ME—like to joke a lot. I inserted Simon's phrase ["outwith"] in several places just because of the reaction I knew it would draw from Keith. Simon did have a lot to do with the Code, but this phrase was just a break from the work." Comments on Chapter 6 (September 23, 2003). So, according to Gotterbarn, it was his joke, not Rogerson's. This joke was the one piece of evidence I came across that anyone but Gotterbarn could change the text on his own, even as a joke.

³³ “The way email is processed in packets explains the other phenomena below. Again I tried to treat the frenzy caused by lack of knowledge about packets in a light hearted way. So I described the most extreme example and then suggested we move on and not further crowd the bandwidth.” Comments on Chapter 6 (September 23, 2003)

³⁴ Actually, all addresses with “weil” were Vivian Weil’s. I had a series of addresses with “csep” (or the capitalized version) before the “@”, because I wanted CSEP’s secretary to receive and print my email. My addresses were: csep@charlie.cns.iit.edu (1996-97); csep@charlie.acc.iit.edu (1995-96); and csep@IIT/VAX.BITNET (1994-95). The old addresses forwarded to the new addresses for a time, adding to the confusion, especially since IIT generally “migrated” addresses in batches, not all at once, and “csep”, being near the beginning of the alphabet would be “migrated” well before “weil” near the end. What was happening at IIT was, of course, happening elsewhere as well. Within a month or two of the day on which Gotterbarn first put his list together, it was probably out of date and, no matter how hard he tried, it was probably never up to date again. Cutting-edge technology cuts both ways (making a job both easier and harder).

³⁵ Gotterbarn considers this email a “an explicit request for comment” (Comments on Chapter 7, September 21, 2005). I do not. The request is, of course, not addressed to me directly, but to Weil (since it was part of email attempted to get me back on the mailing list); it came after Version 2 was complete (and as part of attempting to send it out for generally comment); and it lacked the specific questions Gotterbarn had addressed to members of his executive committee or DMU’s inner circle. These considerations do not excuse me from commenting; they just address the question of whether was asked to comment (and on what).

³⁶ If Gotterbarn’s transcription of my address is correct, the problem was simple but easy to miss, the confusion of “iit” (correct) with “itt” (incorrect). In general, email addresses require a perfection that postal addresses do not. In addition, during these years, email addresses changed much more often than postal addresses because they contained information about specific servers (“charlie” or “VAX”), administrative departments (“cns” or “acc”), and program (“BITNET”) now generally omitted. Maintaining the address list turned out to be a permanent undertaking more demanding than keeping a paper list current. So, for example, something like the February 12 exchange occurred again on: March 27, 1997 (Gotterbarn to me, checking to see whether I got what he sent); April 7, 1997 (from Gotterbarn to me, “you have been added to the PRFCMP-L mailing list”—with detailed instructions); April 24, 1997 (from Gotterbarn asking for name, address, e-mail, and so on to “update database”); and April 29, 1997 (from me to Gotterbarn, thanking Gotterbarn for putting him back on the list but reporting that Weil, weil@charlie.cns.iit.edu, was now off).

³⁷ Gotterbarn never explains the switch from “Introduction” to “Preamble”. I had chosen “Introduction” as a term less pompous than “Preamble” (and in keeping with Mechler’s initial choice).

³⁸ The error is just one indication that Gotterbarn had prepared this memo in haste. Another is his description later in the paragraph of a preamble not yet written as if it were included in the

document sent: “The code also differs from other codes by clearly addressing, in the preamble, the difference between the three levels of professional obligation and organizing the clauses under each principle around these three levels.” The code never was so organized (and the preceding paragraph had admitted as much). Gotterbarn’s thinking seems to have got a little ahead of his editing. That is not surprising given how much he did in his first five weeks in England. What is important, if not surprising, is how his thinking about the Code seems to have been evolving as theories espoused for years met the practical difficulties of writing a code.

³⁹ http://www.iit.edu/departments/csep/PublicWWW/codes/coe/Australia_Code.html.

⁴⁰ Gotterbarn gives no reference for this statistical claim. My random examination of a few computer ethics texts failed to confirm it. Gotterbarn’s Comments on Chapter 6 (September 23, 2003) responded to this observation: “You claim that your readings fail to find any evidence for this claim is telling- Try reading the most influential book written on Computer Ethics- in the third edition now- ‘Computer Ethics’ by Deborah Johnson- the point about professional relations is in all three editions.” I then checked Deborah G. Johnson, *Computer Ethics* (Prentice-Hall: Edgewood Cliffs, New Jersey, 1985). What I found was a book with six chapters not much different from other computer ethics texts I checked. The chapter titles were: Ethical Theory; Professional Ethics; Liability for Malfunctions of Computer Programs; Computers and Privacy; Computers and Power; and Ownership of Computer Programs. The text was not organized by relationships. However, Johnson’s chapter on professional ethics does contain a section, pp. 26-30, on “Professional Relationships” subdivided into four “relationships” (Employer-Employee; Client or Consumer-Professional; Professional-Professional Relationships; and Society-Professional). There is no claim there (or elsewhere in the book) that these relationships cover all the topics of professionalism or even that they are useful as a mnemonic. Johnson makes no pedagogical claim whatever there. So, it seems, Gotterbarn still lacks authority for his statistical claim. And, in any case, the software engineering code is, as explained above, only partly organized according to “relationships” (at least, relationships of the sort Johnson distinguishes).

⁴¹ Or, at least, is true if we *ignore* the intended effect of the premising “In particular” clause. In many, perhaps most, instances, the conduct intended in Version 1 and Version 2 was probably much the same. The problem was finding the best words to achieve that effect (with only our knowledge of English to guide us). A “user lab” would have been a great help in preparing this code of ethics (or any other).

⁴² “Level [or type] 1 obligations” were those that applied to humans as such (that is, moral obligations). Why they should be more, rather than less, aspirational is not stated. *Generally*, we consider moral obligations (the universal ones) to preempt merely professional ones. That “generally” indicates some controversy concerning whether professional obligations may (occasionally) preempt moral ones. See, for example, two advocates of professional obligations as preempting ordinary morality: Benjamin Freedman, “A Meta-Ethics for Professional Morality,” *Ethics* 89 (October 1978): 1-19; and Alan Goldman, *The Moral Foundations of Professional Ethics* (Rowman and Littlefield: Totowa, NJ, 1979). For some important criticism of this “separatist thesis”, see: Paul R. Camenisch, “On Being a Professional, Morally Speaking,” in *Moral Responsibility and the Professions*, eds. Bernard Baumrin and Benjamin Freedman (New York:

Haven Publications, 1983), pp. 42-60; Alan Gewirth, "Professional Ethics: The Separatist Thesis," *Ethics* 96 (January 1986): 282-300; and Michael Davis, "The Moral Authority of a Professional Code", *Authority Revisited: NOMOS XXIX* (New York University Press: New York, 1987), pp. 302-337.

⁴³ October 6, 1996 (Mechler to Cabrera). Shaw had objected to the original 2.02: "This is too arbitrary. Why must their belief be well documented? Doesn't this depend on the intended use of the software? Doesn't 'safety' also depend on the intended use?" Mechler's response had been to repeat what he had said in response to a similar comment concerning 1.14: "Ethics is a set of principles or values that guide our judgments." He wanted to guide the judgment of software engineers toward using all appropriate tests, not just some, and to doing a good job of documenting their beliefs (and what they knew). What tests are appropriate and what is a good job of documenting may depend on intended use and is certainly a matter of judgment (among other things), but what Version 1 said, in effect, is: *Don't do without adequate documentation of your belief that the product is safe, meets specifications, and has passed all the tests you consider appropriate*. What objection could there be to that requirement?

⁴⁴ The principle is "questionable" here (and in interpretation of codes of ethics in general) but not everywhere. It is a standard principle in the criminal law ("No crime without a statute"). The principle is questionable in interpreting codes of ethics because a decent person does not try to do the minimum she can (without actually doing something wrong). The only way to be reasonably sure of not falling below the minimum is to resolve reasonable doubts in favor of the more demanding interpretation, not the less demanding. In ethics, there is something wrong with "sailing close to the wind".

⁴⁵ "[N]o—we put a much more limited definition on this." Comments on Chapter 6 (September 23, 2003). When I asked Gotterbarn what interpretation (and whether "this" meant "maintaining independent professional judgment", "conflict of interest", or 3.06), I got no answer. As the author, I would have like to know. Plainly, I had somehow failed. I would have liked to know how. But time has probably closed the door to such knowledge.

⁴⁶ The conclusion to draw here is that it would be good to have the knowledge that Gotterbarn, Miller, and Rogerson claimed but did not have and good too to admit that they did not have it.

⁴⁷ The (British) "whilst" suggests that Rogerson wrote this comment. Gotterbarn, however, disagrees: "Simon did have a lot to do with the Code, but his phrase ["outwith"] was just a break from the work. Just as My use of the word whilst—whilst I am in England....A hunch—but Don may be funny here because the OED was given as the Source rather than Webster. I actually vaguely recall Duncan [Langford] getting into this issue." Comments on Chapter 6 (September 23, 2003). The crucial email from Langford, if there was any ("vaguely recall"), seems not to have survived.

⁴⁸ See, for example, *Webster's New Twentieth Century Dictionary of the English Language*, 2d ed. (World Publishing Company: Cleveland, 1960), p. 606; or *Thorndike-Barnhart*

Comprehensive Desk Dictionary I (Doubleday & Company: Garden City, New York, 1958), p. 275. If these references seem too old, going to the current Merriam Webster website yields much the same: “to make sure, certain, or safe: **GUARANTEE**. **synonyms** ENSURE, INSURE, ASSURE, SECURE mean to make a thing or person sure. ENSURE, INSURE, and ASSURE are interchangeable in many contexts where they indicate the making certain or inevitable of an outcome, but INSURE sometimes stresses the taking of necessary measures beforehand, and ASSURE distinctively implies the removal of doubt and suspense from a person's mind. SECURE implies action taken to guard against attack or loss.”

⁴⁹ For example, while the OED's first non-obsolete definition of “assure” is “to make safe *against* or *from* (*of obs.*) risks; to insure, *esp.* on mod. usage **to assure life**: to secure the payment if a specified sum in the event of death”, the first non-obsolete definition of “ensure” is: “to pledge one's credit to (a person); to tell (a person) confidently *that* (something is true).”

⁵⁰ Of course, “compensation” carries this meaning in the US as well; the question is whether “Europeans” (or, at least, the British) would also understand that the sense intended is not payment for wrong but payment for work. A visit to the OED reveals that while “compensation” in the sense of salary is an Americanism (2d), the British do use “compensation” (2a) to mean, “That which is given in recompense, an equivalent rendered, remuneration, amends.” So, if OED offers a good description of British usage, and even if Europeans still learned British English rather than American English, “compensation” should have caused “the Europeans” no trouble.

⁵¹ Gotterbarn\Version1\CMTFULG. I had considered having a similar section early in the preparation of Version 1 (with the keyword “Supervisors”) but had given up because there were then too few clauses specifically dealing with supervision. Instead, I had treated supervision as just one of the activities of “colleagues”. Once I had the seven principles, I did not notice that the suggestions coming in had swelled the clauses under Principle 6 to the point that division into two principles, one concerned with colleagues strictly so called and one concerned with software engineers working as supervisors, had become not only possible but desirable. Not then having the benefit of Fulghum's suggestion, I had missed an opportunity to make the code substantially more “user-friendly”—as did the rest of SEEPP/E.

⁵² Version 2 was, of course, in part a response to outside comment, but that is not the same as having the new version put to the test of outside comment. No code writer knows whether a provision will pass muster until it has. Every change in a code, even the change of punctuation, is a bet that others will see in it what the author does—or, at least, not see anything to object to. Just as the authors of Version 1 lost many of their bets, so the authors of Version 2 would lose many of theirs.

⁵³ Gotterbarn asks: “Why late in the game—this was a draft that had not passed muster from the community it would serve?” Comments on Chapter 6 (September 23, 2003). We have here a question about how to interpret the set of facts we have before us (Chapter 6). I say “late in the game” because (technically) “the game” was already in over time. SEEPP had already missed the original deadline for completion of the code, and was now working beyond the second deadline.

(That the “game” in fact went on another three years does not change these facts—or the perception they should have produced at the time.) *My* reading of the comments of the community the code was to serve had been quite positive, except for those of Failey and Shaw. And, on the whole, the criticism of Frailey and Shaw seems to be more a matter of misreading the code, or simple ignorance of other codes, than hostility to what the code actually said. What constitutes “passing muster” here is a political question, not a factual one. Gotterbarn may well be right about what was necessary to get the Joint Steering Committee to pass the code, any code. But I have no idea how to decide that he was right. And I am pretty sure he doesn’t either. So, the best I can do here is set out the alternative interpretations.

⁵⁴ Gotterbarn thinks this explanation of his conduct in the preceding two years to be “nonsense”. In particular, he pointed out in comments on an earlier draft of this chapter: “What was going on was not the engineering method—it had nothing to do with large projects. I have managed long-term multi-person projects. The problem was the documentation and review method and the way it was being given lip service.” Gotterbarn Chapter6cmt (October 31, 2004). Gotterbarn now has no more access to his thinking in late 1994 than I do. We are both involved in reconstructing it from the evidence now available. One problem with his reconstruction is that it leaves unexplained why he initially went along with IEEE approach instead of insisting on the ACM approach or working out a compromise, and why he later changed, exactly what my reconstruction does explain (and what it set out to explain). Gotterbarn may be right that my reconstruction is wrong but, as far as I can tell, the evidence makes my reconstruction the more plausible. IEEE procedures got a good deal more than “lip service” 1994-96. The CFP, Operations Guide, and SEEPP organization are “body service” (deeds) rather than “lip service” (mere talk).

⁵⁵ Gotterbarn/94-95 misc/Boston 6-94.

⁵⁶ See http://whatis.techtarget.com/definition/0,289893,sid9_gci214112,00.html (September 4, 2004): “Typically, a skunkworks has a small number of members in order to reduce communications overhead. A skunkworks is sometimes used to spearhead a product design that thereafter will be developed according to the usual process. A skunkworks project may be secret.”

Copyright © 2009

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