

Don Gotterbarn Interview

Interviewee: Don Gotterbarn

Interviewer: Tom Misa

Location: Illinois Institute of Technology Main campus

September 22, 2005

© Don Gotterbarn

Contents:

- Preliminary comments
- Questions about the Nature of Software Engineering and Professions
- Biographical Questions
- Development of the Software Engineering Code

(Comments marked by a * were added by Dr. Gotterbarn to clarify points made in the 2005 interview.)

Preliminary Comments:

* Commitment overcame management problems

DG: Looking back over the development of the Software Engineering Code of Ethics and Professional practice you are amazed by the number of disasters, pitfalls, traps that none of us anticipated. And what happened was it looks like a project that was done from the top down. At an International Software Engineering Conference in Baltimore there were several groups holding concurrent meetings. All were considering a suggestion emailed by Fletcher Buckley recommending [a code of ethics]. The IEEE Board of governors appointed a Blue Ribbon Committee to investigate the recommendation and the issue of professionalizing software engineering. There are common characteristics of a Blue Ribbon committee; all of its members are recognized in the sectors, are incredibly busy, but may not know one another and have very different views about the nature of software engineering and what is best for it. What ultimately happened was that the abstract direction from the top layers of Blue Ribbon Committees and Professional Organizations came down through multiple layers to a group of volunteers with equally diverse and with wildly strange backgrounds and commitments that managed to produce the final accepted version. It still amazes me how this very significant, effective document that's had an impact on software engineering already survived the process of its generation.

The point is that everybody was making mistakes all over the place and a certain stick-to-itiveness of you're committed to this and you're going to bounce against this hurdle and keep going and keep trying, brought it through. And it's amazing because the number of people and the pieces they played is just exciting when you look at it. And you can't give credit to all of the key players, that is one point. This is an event that happened in spite of; let me say in the strongest sense, in spite of a whole lot of stupid mistakes and contentious things going on at within and between the various layers of managing this

project. It still came out as a successful project. The commitment of those involved overcame these problems. So, that's interesting and exciting. That's one.

Is that you just turning it off and saving the tape?

TM: No, the tape's running. Maybe we could say this was Don Gotterbarn's spirited preamble. We're doing a series of oral histories from now till doomsday. It's, let's see, if I get the date correct here, September 22, 2005. We're on the IIT's campus. My name is Tom Misa. Don has some preliminary comments and I'll shift to a set of questions but go ahead.

* Overconfidence in technology led to misunderstandings about the progress of the project

DG: Well there is a second comment, sometimes people put mistaken faith in technology and it's interesting that technologists do this. So, given a horrendous state of email at this time (1993-1999) emails were lost or corrupted, work had to be redone, some people came in and out of the project because of lost contact or changing email addresses. This project spans the development of the internet. And all sorts of strange things happened, one because of the inadequacy of some email equipment, two because of the incomplete understanding of the way email worked by some of the technical practitioners who participated in this. This generated all sorts of misunderstandings about human contributions.

What's really interesting to me about this is that the faith was that the technology is okay, and it was the human beings who were evil, rather than looking specifically for the technological issues. That kind of confusion permeates the current text written by Michael Davis written about this. That's point two.

TM: Point two, okay.

* Mistaken belief that public documents accurately reflect the underlying character of a project.

DG: Point three, when writing history, basing that history on the existence of public documents that are shared in a very public way, basing all inferences on that is in fact a considerable mistake.

Questions about the Nature of Software Engineering and Professions

TM: So now I'd like to ask you a couple, a set of questions really, that deal with the field of software engineering and also the notion of a profession.

1. Software engineering as a term seems everybody points to this NATO conference in 1968 as an origin point. Could you give me suggestions of other names that would occur at the time to describe activities that became known as software engineering?

DG: Computer programming or other names that could serve as alternatives to software engineering. In '68 you're still relatively early in the history of computers and the history of computer systems. Computer systems at that point primarily were accounting systems, bank tracking systems, government types of systems; systems which fired missiles, large companies are getting interested in it. At that stage of the growth of computing you really have no discipline. People were figuring out what they were doing and not recording it. The kinds of catastrophes or examples of flawed software that occurred, led some people to lock onto the word "software engineering". The whole point of that was to indicate that you needed a disciplined approach to the way in which you developed software.

So those people who uttered that at that point did not have in mind licensing regulations in the United States; did not have in mind registration. All of this has become a distraction because of the use of the word 'engineering'. In fact, in most other countries the word "engineering" does not carry the honorific tone that it does in the US. Using this word when you bring it into the US causes all sorts of difficulty, because being an 'engineer' in the United States is a protected title. And I have to be careful here because I don't have the exact factual data but in general I believe in order to call yourself an engineer you have to have an engineering degree and graduate from some college that has a certified degree.

TM: It's typically an ABET accredited degree and we actually have issues because people would like to talk about information architecture and information engineering. We can do one but when we say 'engineering' it has a special loaded term.

DG: In every place else in the world they are now calling these folks ICT professionals; information communications technology professionals. It's not software engineer. The issues of the kinds of craziness that came about just because of the choice of the word "engineer" in software engineering are evident, I believe it is in the state of Indiana or Illinois, where in order to do anything for the state government in terms of software, you must have graduated with an engineering degree from a ABET accredited college. The last time I looked I believe MIT is not an ABET accredited college. So you can have a computer science graduate from MIT, and someone who had graduated from some small state college with an engineering degree having had no computing, and the job would not go to the MIT graduate because they were not a quote "engineer". So the word "engineering" is a problem but its use at the NATO Conference was a label that says, "Here's a goal we have in mind when we do things."

TM: You mentioned the different failures that were evident in the 1960s and the failures would have been, the one that would have been on my mind, would have been the software written for the operating system of the IBM 360. Was that one of the examples of failure that people have in mind?

DG: Well, that's one of the types of failures. But interestingly enough, the failures at that point were not catastrophic failures but financial failures. But I tend to talk a lot about catastrophic failures because it gets people's attention quickly, and you don't need a long budgetary explanation. But most of those tended to be software delivered late, software

which would not do the job, sometimes misunderstandings about what was going on. But you now have this label ‘ software engineer’ that probably was a tremendous mistake. To simply talk about professional software developer would be better than the label “software engineer”, because there would be no discussions about engineering or whether you’re doing engineering ethics.

And that brings me to your second question, which was one about what I consider to be a professional.

2. Defining a profession and being a professional

TM: Well, there are two things. How would you define the profession and then what does it mean to be a professional?

DG: Okay, they’re different questions.

TM: Two different questions, correct.

DG: And in fact the question “How do you define a profession?” can be answered two ways. One is fairly simple. Is it recognized by law as a profession? So the Taft Hartly Act in the 1970s did not recognize computer developers as professionals. And some very strange legal decisions are based on that, for example people who we would now call crooks and villains were exonerated because developing computer software was not legally considered a profession. The other approach is the philosopher way of defining ‘profession’ by setting down a set of criteria as to what constitutes a profession. These criteria vary from philosopher to philosopher. They generally include things like there’s a specialized body of knowledge. Now it isn’t simply a specialized body of knowledge because plumbing is a specialized body of knowledge, carpentry is a specialized body of knowledge, but it’s a body of knowledge that requires a higher degree of education. So that’s the body of knowledge part.

Another part is that they will apply the skill in the service of society. So lawyers and doctors are considered professionals because what lawyers do presumably helps society to function more smoothly, what doctors do is to care for the health of society. People talk about journalism ethics and professional journalists in the same regard. And I am going to resist making any side remarks about this.

TM: Don’t worry. No Enron jokes... But it’s important to say that those are goals and ideals, not everybody lives up to those.

DG: And they’re targets, you set. Now incorporated in this is one of the things that causes some difficulty. Professions “have a professional organization representing them.” So in the US we point to the American Bar Association or the American Medical Association, and then they have a set of standards by which the people in this profession apply, aspire to, and try to follow. And your adherence to these standards is part of your right to call yourself a member of that profession. Now there are a number of problems. When we talk about software engineering being a profession and trying to professionalize

it, one of the early concerns about this was the claim, “Well there is no standard body of knowledge. Software engineers do too many different kinds of things.” And the people who say that talk from software engineering as if this made it different from other disciplines. But when we look at medicine as a profession it includes the dermatologists, the oncologists, the optometrists, the ophthalmologists, the endocrinologists, the...and I’m running out of labels for now. But each of those...

TM: All doctors for all different body parts.

DM: All doctors for all different body parts and then there the holistic ones who do the whole body. And so that business about there’s a wide diversity of software development does not make software development different from other professions. And then there is an argument, some of the philosophers argue in their books that, “Well there is no single representative organization for software engineers or computer folks.” And the presumption is there is a single organization for medical doctors, but that’s incorrect because I can go to Tonga and practice medicine. I can join the Tongan’s Medical Association which has a different set of standards then the American Medical Association. I think that the important thing is not the appeal to the organization but the common beliefs and commitments to a way in which medicine is practiced. Don’t cause harm to your patient. Apply the best techniques you can for the health of your patient. Those kinds of principles can be generalized to most things that intend to be professions. What we talk about when we talk about a profession as far as I understand it, must include something called ‘a higher order of care’. That is because of the power and impact that you have, you have more responsibility to the people, to your patients, to your customers, clients, whatever they’re called in your profession.

TM: Could you give a description or a definition of software engineering? I’m trying to recognize that it has many different overlaps but what would be the core problem, or core method, or core insight that someone who’s doing software engineering would have that someone who’s doing computer programming or system’s analysis might not have?

DG: I think software engineering has to include both the general and the particular; an understanding of the way systems operate and what you’re doing in the system and how what you’re doing will impact the application of that system and the development of that system and the way that is used in society. So a programmer who’s writing a piece of material can understand the system view. I view software engineers as people who take a systems view to their development, even though because they are developing a small piece of software, you might call that person a programmer. Let me distinguish programmers and coders, or people that I might be willing to call software engineers who primarily write code. In some industries you are told this is what you will do, this is what you will code, this is the way you will code it, and this is what our standards are. We can actually write computer programs that comply with this. People who merely do this I think of as ‘coders’. They’re not software engineers. They don’t respond to, or realize the moral responsibility, or the way in which their systems relate to other systems. They don’t understand what’s going on and use that to control the way in which they do things.

Here's a very technical example. One of the ways in which you can write a piece of code to repeats things is to write a statement that says, "Do this while such and such is true." Another way you can do it is to say, "Do this until such and such comes about." Most programmers, most coders treat those as identical. Someone who has a software engineering approach would understand and take into account that, if they use this and they're writing a program that dispensed a very powerful medicine into a patient automatically, if they use the "do while something is true" they will prevent overdosing the patient. If they use the loop that says, "do this until something is true," they will have already overdosed the patient and then found out they've made a mistake. Having a systems view and approach to this is what I consider to be part of software engineering or a major part of it; to understand these issues. It's not just technical skill.

TM: The potential moral consequences, that is the consequences for the safety, health, welfare of society, anybody who's using the program or if it's software that's overseeing missiles, or nuclear weapons, or nuclear power plants, or chemical plants that's everybody in society.

DG: What I've done... The answer is, "Yes, but." What I've done is that I've fallen into the trap of using my short hand examples which are always life threatening ones. In the same regard, writing a program for a hospital billing system where you don't understand what you've done and the hospital goes out of business because your program malfunctioned or because you didn't write the program right, or writing a system that means that people won't buy a product from a certain company just because it's hard to use and you didn't develop the user interface properly are examples of it as well. Writing a web interface and using Times Roman font and pure black letters on a bright white background is also a mistake that the systems developer, software engineer should know about, because that kind of interface will give a dyslexic a headache trying to translate it because of the serifs of the font against the bright white background. And it actually is painful. You've created a website that causes pain.

TM: Which on the doctors' Hippocratic Oath that's one thing you're not supposed to do, cause harm unneeded.

DG: And I have to be... I'm using the word harm I don't just mean physical harm but I get people's attention when I talk about physical harm. It's a weakness in my approach though because they think I'm only talking about critical safety stuff.

And the software engineer, generally, people will talk about the software engineer as being involved in large systems. I think large is a relative term and you could have a software engineer applying the principles of good design and good development even in small systems.

TM: Could you say something more about the notion of the system's approach in my own field history of technology that a particularly...that we have a specific definition of a systems' approach? I'm not going to give here because I'd like to hear yours. Maybe we

can talk back and forth about it but I'm interested if they overlap. A systems approach as opposed to a different kind of approach?

DG: The way I've been using it here involves I would think two basic elements. One is the systems' approach involves you thinking about the place of what you're doing in a system and its relationship to the whole system and how that might change it. So even if you're doing a small part of the system, you can still take a systems' approach to that. The other version of the systems' approach is you're developing the whole system and you're worried about the impacts of the system outside of its what we call side effects, the side effects, the direct effects and side effects of the system you're developing. And how the way you're developing it will change those direct effects and side effects.

TM: So the direct effects are ones that are intended and the side effects may not be intended but are just as real. Unintended consequences, consequences real that have to be contended with.

DG: Several examples... In New Zealand last year there was some modification made to a device that's used to do cataract surgery. And the technician who was modifying the device noticed that the functionality of the off button, which is retracting devices from the eye and turning off the system and so on, were exactly similar to the functionality of the emergency button, retracting the devices and so on. So he simplified the code and made it more consistent by removing the emergency button from the left hand side of this device and all functionality is in the off button. Otherwise the device looked about the same, and of course you can anticipate immediately the consequence. There was a situation where a doctor in New Zealand was working on someone and the emergency came up and the doctor hit the left hand side of the device where the emergency button had been located, so it wouldn't be hit by accident, and didn't feel anything. And the doctor reacted urgently and slammed the device again with his hand where that button should have been, because the device was doing bad things to this person's eye, and actually moved the device while it was inside the person's eye.

TM: And...failing to take a systems' view. Seeing the object...

DG: Don't use the word "seeing" in this example.

TM: Seeing, that's right. Perceiving, accurately perceiving, in fact, and acting on that. My own field uses systems in the first view; thinking about the whole system even if you're working on a small invention that has consequences for a water, electrical system or production system. We typically don't take that second view about consciously thinking about the impacts. Although, that's an important thing when you're doing technology assessment in a whole system is to think broadly about that. And that's a nice example.

DG: Let me give you one other example that will show that quite simply. Again these are horrifying types of examples. In New Zealand, when I was there, someone took one of these devices that you use to remotely unlock your car, and had this wonderful idea about

programming to, when you press the button, start the car remotely. Now in cold climates that's good because you can get the heater started as you're walking up to your car and things like that. In dangerous environments that's good because you can get the car started and run into your car and run away if somebody's chasing you and you have your car parked in an environment that you're uncomfortable in. In New Zealand, which is a highly volcanically originated country, most of it consists of very steep hills and so on. In the city of Auckland, which is built on seven volcanoes, there are almost no level streets. This is where the program was instituted. Now if you know about driving in New Zealand, they use standard shift. When you park a car in a standard shift you normally do something, even if you forget to put the brake on, you put the car into gear. And the obvious and expected thing happened. Shortly after the remote start was introduced a pregnant lady was walking in front of a parked car, a gentleman while approaching this car started it remotely. The car jumped and crushed the lady between the car in front and his car. She went to the hospital and people minimized the results saying, "Oh well, it was just a 10 week premature birth and they're both okay." And according to my limited medical knowledge, a 10 week premature baby will have trouble for the rest of their life. The simple notion was the person who was writing this, it worked; you pushed the button, you started the car. You should've taken into account that there are standard shift cars, and could have done one simple thing and made it so it just doesn't work if your car is standard shift, or it has a fail on it that if your car is in gear and you have a standard shift it works. Ten minutes more work in the design would have changed the impact of this whole thing. The programmers did exactly what they were told, but they were coders. They didn't code well.

TM: But then you would have the expectation that a software engineer would take a broad enough view. That is, not looking at the narrow technical functionality of the code but also the broader social - you could have cultural and economic - impacts and to take those impacts into account while they were doing the design.

DG: That's what we mean...Yes and that's my interpretation of a higher order of care. Making software engineering a professional computerist, if you want to call them that. It's just like people who are doctors could technically do something like, if you went into the doctor and said, "Doctor I want you cut off both my arms at the elbow tomorrow," the doctor could say, "Yes," and do that quite successfully and effectively, and make money. But you and anyone else would deny that that doctor was being a professional because they didn't take...there may be some reasons but if the doctor didn't ask and explore, that doctor's not doing their higher order of care. That's what I mean by a professional software developer. And I get caught in all of the rhetoric that goes around the word "software engineer" and "software engineering professional" and so on but you have in these words gotten my definition of professional and professionalism. And now to be certified as part of your profession you need all sorts of other standards and right now software engineering doesn't have that, in part because of all the competition with the engineering profession. The engineering profession wants to impose engineering standards on becoming a software engineer and so you must do things in some cases, studying, I've forgotten it, the one with heat transfer, thermo...

TM: Thermodynamics.

DG: Thermodynamics, you must have a course in thermodynamics. Well, most software engineers don't need that but to get that label "engineer", there are issues about that and that's why. The word "engineering" in software engineering has caused a problem. Calling this code of ethics a software engineering code of ethics in one sense causes a problem. We built a code of ethics that will fit professional software developers, which is what I think most of us who were on that committee had in his mind as software engineers. And it's only the 'engineering' word you will see when you look at the history of the effort also caused difficulty or facilitated certain sorts of criticisms because people could then say, "Well this doesn't look like an engineering code." Well if it did it wouldn't look like a software code. So it has some engineering elements and some software elements and so on.

TM: You've defined software engineering as having a systems approach, a certain kind of attention to the whole system, an up front attention to the downstream consequences. And I'm wondering whether there are any special methods that were important as software engineering evolved, especially in these early years say between the 1968 NATO conference and the founding about 20 years later...? There's kind of a funny lag between the early interest in software engineering and the founding in 1985 of the founding of the Software Engineering Institute at Carnegie Mellon, which was an early institutionalization of software engineering as an ongoing concern.

DG: Some people think that labels make reality. So if you call a law that favors people who do pollution a 'Blue Skies' law you'll think that, "Okay now we're paying attention to it." The software engineering label came out and it was used at the NATO conference. The Software Engineering Institute was in response to demands by the military for good software. The Software Engineering Institute, when I was working in Pennsylvania at the time of its development and it was funded by the Air Force primarily. Because they were getting unreliable software, software that would fail at certain points, software that would not run at certain points, and when you're flying an airplane you consider it important that the software that controls the airplane continues to function properly.

The funding of the Software Engineering Institute in 1985 did not indicate a gap in the concern for the development of good software. We've been worrying about it and doing the things like that, working on the development of good software. There's an inference that people stopped caring between '68 and '86. That's nonsense to think that no one is looking at the development of the way in which you write software. You have computer science becoming a discipline and [it] is taught in universities. You have people writing textbooks about the way you ought to do testing and what the limits of testing are. You have people designing techniques that say, "If you're developing safety critical software, we can apply mathematical criteria to it," called mutation testing, "which will allow me to give you a number applied to the degree of reliability," and we can do these things. One of the problems is, as the education was developing you still had a community that came into developing software without going through this education. They figured out how to do it on their own, and thinking because they could make a program work, that it

was okay rather than applying quality standards to it. So, in part what the Software Engineering Institute intended by its label is to put up front again. “These are standards that we will set up, and we will talk about standards, and we will disseminate these standards.” It was not, “Oh! Golly, we’re making mistakes. Let’s pay attention to it again after 17 years.” To characterize that period is wrong. I could waste the whole day giving you examples of each new and interesting discovery. I mean, we use to have a way of writing code that was called spaghetti code. And you just wrote statements until you got the thing to work. Then people figured out writing software was a form of problem solving and you’d apply strategies to it. Then we actually organized the strategies and developed something called structured programming. You have books by Edsger Dijkstra called The Discipline of Programming, which appeared well before ’86. Which were hallmarks about... and ways in which you could prove programs mathematically. All of these efforts are doing things to improve the process of making software, and applying, and developing standards.

TM: So there were ongoing practices within the computer science discipline and within people in the field?

DG: And people who knew it.

TM: But people didn’t label it necessarily as software engineering.

DG: No.

TM: So, it didn’t have a kind of visibility. When we look now, back and say, “Well what is the history of software engineering?” we would miss those important practices. But if we changed our view and said, “What were people doing to improve the quality or reliability of software?” then we would...

DG: Then you’d trap all that. Yes, that’s correct. So harping on this particular label is a problem.

TM: It’s a particularly powerful label. So if you try to get a picture of what it does and what it doesn’t do seems important. And it still seems a bit of a puzzle in my mind why NATO used that label in 1968. And that’s such an early time within operating systems, within programming that it seems like it’s a curious early usage of that term. And I’m just curious whether you might have reflections on why or where that term came from.

DG: I historically know that it occurred. It had been just part of my general knowledge things, which means it may be false, that the primary intent of using this term at the NATO meeting in Garmish is to declare that we have to move in this direction. Some people think that it was designed to be contentious, rather than aspirational. I tend to think it was aspirational, but then I live in the profession so I tend to give up a positive interpretation of this. The problem in the ‘60s, you’ve got to remember, [is] that the main frame became an entity in maybe 1956 which was by several universit[ies], so when you talk about 1960... I graduated college in 1964 [and] there was no computer science

department in the country in that time. So the business that we have to call attention to the standards, and that we ought to keep developing the standards, using 'engineering' as a label is interesting because the people who do the history of engineering would say, "Those people who used it, and garnished it and made out," misunderstood engineering. They thought engineering to be a fixed and stable and solid discipline where you open the rule book, you have the rule, this is what you do, and you do nothing else. There is no invention, or creativity in engineering. You have the standards and you apply this mathematically model and that's it. And we're in a building that had to violate some of those mathematical models right now, or had to do some variations on what those standards would have been. In Garmish, those people who look at engineers go, "Bridges work. Buildings don't fall down. Software's falling down. We should follow the 'engineering model'."

Hence, software engineering, some of us use phrase like engineered software. I can tell you when I do safety critical systems for certain kinds of things or will have a great impact on society; I carry the engineering levels to a very high principle. That is the standards I know they're applying. I think to worry about [the] Garmish use of the phrase may be interesting historically but the real transitions started to occur as software began to spread. People used to just gripe about the telephone bill, and the electric bill, and the machine that did this. Now they have pacemakers in their chest. They drive their car and their car is mostly computers. Most engines are controlled by computers and not by people. They travel down intersections where the flow of traffic is controlled by computers and scanners and the amount of traffic and calculations. And all of a sudden you have to worry about, "Wait a minute. We have people who are hired by companies who may have had no college education and not know any of these standards about good testing, writing software that affects the airplane I fly." I flew in an airplane here today and fortunately we knew about a thunderstorm ahead of time. Thank you computer. It was a very comfortable ride because the computers directed the airplane to fly above the turbulence. Knowing all of this, you now have these great efficacy things that are going on by computers. And it is no longer acceptable to approach it in a haphazard fashion.

[end of tape 1 side A.]

Biographical Questions

TM: Don, I'd like to shift just a little bit and ask about your biography and your background. People of your generation ended up getting into computing from a variety of different places. You mentioned before that it wasn't an established discipline of either software engineering or computer science. But surely you must have one of the strongest backgrounds of philosophy of anybody that I know doing software engineering, both a bachelors, masters, and a PH.D. in philosophy. A PH.D. in 1971 and then additional technical training in computing. Maybe you could say a few words about what prompted your interest in computing. Where that came from? And how you made the switch from philosophy to computing? And then I have additional follow up questions about how you got interested in software engineering and software engineering ethics. But can you

describe a little bit about your background in philosophy and how you became interested first in computing as a field of practice? It doesn't follow in my mind logically from David Hume.

DG: Everything... There is another National Science Foundation research project on moral exemplars in computing that has me as a subject and it was insightful because of some of the questions I got asked. All of the questions and everything seem to fit if we take into consideration my father had what would be called a craftsman's approach to everything. Everything you did should be thought out, should be organized, you avoid mistakes; it's unacceptable to just jump into things in a silly way. That was driven into me, inspired me. I was also, given that approach, you look at all things as a problem to be solved and to be organized. I enjoyed looking at very large problems. I use to get myself in trouble with large problems because I would pick things that had been proven to be unsolvable and think that I had the ability to solve mathematical equations or do things of that sort. And I would always write papers on the hardest topics. It led me to studies of religious theories and trying to organize and structure large religious systems. Which led in part to the interest in philosophy and the approach was always an empirical, scientific approach. As a student growing up in New York, my high school days were at a special science high school in New York, again the nerd approach.

TM: Which high school was that?

DG: Newtown Technical High School. It's the Borough of Queen's version of the Bronx's High School of Science and Brooklyn Tech and so on. Each borough had its own science high school. So I went to the local science high school. I was going to be a chemist at that point in my career, because I loved the way in which you could add chemicals together and calculate their structure ahead of time and predict certain kinds of relationships. Then some place along the way, I got interested in philosophy as a large structured system again. But my approach was always the empirical science piece. Experiment, test, get evidence, work it out, and develop the system. So lots of my philosophy was spent dealing with logic and writing out logical, structured arguments.

Logical, structured arguments, (jumping way ahead) are very much like computer programs. In a logic argument you set up premises and give evidence for the premises, and you feed those premises into the next set of premises, and you change all those premises by drawing a conclusion and saying, "Here's what the conclusion is." A computer program or a computer system takes series of inputs, massages them throughout the program; in COBOL they are even called paragraphs, in paragraphs, and passes the output of that paragraph onto the next paragraph. It's like writing a thesis.

And all of the things I did had a common problem solving methodology, that's me. You give me large difficult problems. So a lot of the systems I worked on were not small systems that fit in your pocket but systems that ran military systems that tracked munitions, database systems for police departments, and things of that sort. That's the general characterization; I've finally come to realize that this characterizes what I do.

Even now in my spare time, I do watercolor painting. And watercolor painting requires incredible up front problem solving because you have to anticipate what will happen when each color is mixed, when you put one color on top of another because there's no correction and going back. And I find it laughable that my spare time, relaxation involves the same process.

TM: This is deeply imbedded in your habits of thought.

DG: I say it even stronger than that. This is me. This is the person I am. Part of the craftsman's model is certain things are not acceptable. And shoddiness, haphazard[ness] is not acceptable.

Now how I got to what I'm now doing. One of the virtues of being a philosopher, and why it fit that mold, was if you wanted to do the philosophy of the relationship of subatomic particles, if you were a good enough philosopher, you could make that fit the subject and actually teach a course in it. One of the things I did there was a philosopher called David Hume, who talked about the way in which you gain knowledge. I particularly liked Hume because he always wanted to point to the evidence and say, "What do you see?" and it fit my mold. And it fit the mold of applying logic to solving problems. He also had some theories about causality and how one thing causes another. We won't go into those here but that was one of my first papers on Hume. While still in graduate school, before I had even written my thesis, I had published a paper on Hume and causality. Now you look at computers, and when people in the early days started talking about the way computers worked and the computers as if they made decisions and had some kind of causal impact, I wanted to look at what was going on there. And so I began to look at computers, and read about them, and follow some material, and sit in on some classes at various universities. One of the great privileges of being, one of the great pleasures, about being a college professor is your colleagues will let you sit in on their classes and learn things from them in your spare time, and if you get cut short and have to go back to work, they don't object. So I was learning things about computing.

TM: Did you have an early experience with a particular computer or computer system? Or was it more a general interest in systems of knowledge and types of evidence and the kind of logical...

DG: Systems of knowledge and types of evidence...

TM: ...Rather than a piece of hardware that you may have had in high school?

DG: No. High school for me was 1956 to, no 1957 to 1960. The piece of hardware we had at that point was the slide rule. There were no calculators at that point. Just as people carry their laptops to technical schools now as marks of intelligence, when we rode the trains to school you knew who went to Tech because we had our slide rule cases hanging on our sides as badges of, "We are the elite technicians. We are..." they didn't have a word 'nerd' at the time, but we were the technical nerds of that time but the engineers of society.

So the answer about the hardware is that the hardware actually came late in the game, only playing a little bit with some of the hardware at the various schools. We're talking schools having DEC PDP 11 hardware that had 128 K in a mainframe that ran the university system. I'm talking 1970. PCs were not invented yet. The only hardware you could get to was mainframe computer stuff and of course since it carried university data that access was very limited and guarded. You could not play with the machine, so to speak. And people did not have, and I was not interested in the hardware even there were early PCs like Altair. I liked the large problems and addressing them.

In philosophy, I was doing all right. No, it was a success. And I left the place...

TM: In your professional career...

DG: In my professional career as a philosopher.

TM: Making publications and stuff.

DG: I started at Wichita State University and they normally had a 7 year run to tenure, and they were going to give me tenure in the fourth year to keep me there. And then I got a job offer from the University of Southern California to go to their School of Philosophy. I, not knowing much about the work world, went to the University of Southern California and encountered an environment that I found uncomfortable, because it was a political nightmare and you could not, if you tried to be a fully ethical human being, avoid difficulties. So I said, "I have [had] enough of this." And there were other things going on with the family so I went back to Pennsylvania, I went to stay with family in Pennsylvania, my wife's family.

TM: Just to review those years you were at Wichita State during 1971 to '74 and then you went to USC, the School of Philosophy...

DG: '74 to '76, I guess.

TM: You were there either two or three years...

DG: Two years.

TM: ...and you left to return to Harrisburg.

DG: And in having...

TM: You felt the environment at USC was very difficult politicized, or unethical in just some other way?

DG: Politicized and there was some misrepresentation about the job that they were offering me. That is, I was going to have a certain leadership role and a certain function and... Well that's enough said about that.

TM: Okay. It was a combination of both, you'd say, family interests as well as professional dissatisfaction.

DG: Professional dissatisfaction and the family interests, as you said. Also, my wife didn't care for Los Angeles. We left Wichita where we were very comfortable, had some good friends. The Wichita philosophy department was quite aggressive in doing its philosophy and a quality department. After leaving, now you're in this environment that's not comfortable for my wife, and not comfortable for me, and it wasn't professionally satisfying. I quit the University of Southern California without having a job in hand. And went to Harrisburg, which was a bad political move and a bad employment move.

TM: Something of a risk I'd say.

DG: It had its bad and good points. It was time for one of those life decisions of, "What shall I do now." And computers are getting more into the forefront. So you say, "Well, I can do that." And that was all it took then. That's all it takes now. You want to say, "I do software development," get some paint, write get a sign, hang it outside your door, and you now are certified. Well not certified, but that's what it takes. And so I did some...

TM: You had auditing classes, both at Wichita State and USC?

DG: Just USC.

TM: Just USC?

DG: Just USC. So by personal contacts and things you make some contacts with people. And you find that philosophy was wonderful problem solving training because you'd sit and talk with a businessman about what they needed, what they wanted in a system. And they'd say, "Well I have this accounting system that does such and such." And you'd just listen to them describe their system, doing a logical analysis in your head of the system and realize that they're taking a very circumspect route to do something simple. And they had taken that route because it had always been done that way. And you would simply say to them, "Well, is there a reason you don't do it this way?" And they would think I was a business genius, which I always found humorous because I hadn't had any business courses. And it was just the structure of their system that I would reorganize in my head while listening. And you would get a job. And they would be so pleased; they would recommend you to other people. But some of the work I wanted to do required that I have some credentials. So I went to the local community college. And in my spare time, while, I spoke about negative time. During that period of time, I was...

TM: Negative time is that time you can't put on the calendar but you squeeze out here and there. In the evenings, early in the morning?

DG: Right. So the time consisted of teaching a course at Dickenson College as a professor, because one of the faculty members was on sabbatical they knew my record. That was teaching one course, the course he had to teach.

TM: And it was a philosophy course?

DG: Yes. I also taught a philosophy course at Harrisburg Area College because I was a cheap commodity and available. While taking classes in the data processing program, while working for other people, also raising a family and coaching soccer and other things, so I think they all fit into negative time. And if you ask what I did in those first two years in Harrisburg, it was set up some standards. I went to work and did some work with Blue Cross and worked for them, worked for the federal government actually as a federal government employee for a full year, and then went back out into consulting again. So it was quite a mixed bag.

TM: With Blue Cross you were doing also computer and systems analysis...

DG: Right, systems analysis work

TM: And with the federal government in Harrisburg, that wasn't part of the state but with the federal government?

DG: No it was the United States Navy Spare Parts Control Depot where I actually was an employee of theirs for one year. Training systems analysts, working on...working on some weapons systems, and doing work with nuclear submarines. And that's a new set of ethics issues.

TM: You were training systems analysts or doing training and doing analysis of systems?

DG: I was training... I had a teaching function. I would periodically run courses for the Navy at the Navy Yard but my attachment was as a systems analyst to a particular division of the Navy Yard.

TM: You're not talking about the Philadelphia Navy Yard?

DG: No. Later after I left the Navy Depot I had contracts with the Philadelphia Navy Yard while it was still in existence.

TM: This was systems' work and were you aware at all at the time of there being people thinking about all these quality issues as sort of the problem with doing non-systems' analysis of programming? In other words, I'm trying to get a sense of where you've described your interest in logical systems and made a connection to computing, did you see yourself then primarily as somebody who's interested in a special type of programming, this systems' analysis that we would now would understand as software engineering?

DG: I was... Here's now the connection to things I've said earlier. Earlier I had spoken about a haphazard approach to building systems.

TM: Spaghetti code is one term you used.

DG: Spaghetti code, just building systems without thinking of their impacts. The character that I've described to you earlier is Don Gotterbarn, the person who loves to solve complex systems. You can lose any awareness of the world as you get immersed into the complexity of the system. I was that human being. Let me solve the problem. I'll solve the problem you give me and I think I will have done a wonderful job. The experience in Harrisburg was the education of Don Gotterbarn about systems work. Simple examples. Also in my spare time I did voluntary work up at the Camp Hill Public Library as a reference librarian. I had all this education, this academic knowledge which was not getting used in these other areas. And they also knew about my computing work. So one of the evenings the librarian is pulling books off the bookshelf and throwing them on this cart. She's culling books that she believes were not checked out in the past two years. They're going to be disposed of because they have to make room in this physical library for new books. They have to make shelf room. And I look at this woman and I say, "I just read that one last week." And she says, "Oh," and she puts it back. I ask, "You don't have any way of knowing what was checked out in the past two years." "No, but I work on this." I said, "I can write you a program that will do this." Their check out system was computerized and so I wrote a very, what we would call, quick and dirty program. When you checked the book out, it would take the record of checking it out, take that whole record it was about to write and write it to my database. And then we checked that list against the existing list of books and we could see which books were not checked out in two years. A simple program, a wonderful solution., done with the best intentions.

We had some trouble around the library and the police wanted to know what a particular young lad had checked out. And I was there when someone said, "Oh, well we can use Don's program to do that. Can't we Don?" And I realized is that what I had done was to take the complete record; person's name, address, phone number and the book they checked out and recorded that along with the date. And my program only needed the book and the date. And because of the way I had designed that program the police were able to intrude in a way that I thought unacceptable for the reasons that they had.

There are several of those kinds of experiences that I had, both personally and watching others. And you go, "I'm developing software in a haphazard way." I'm seeing others not bother to educate themselves and keep up on the latest technology and changes. They're developing them not just in a haphazard way about impact, but a haphazard way about a system. And this I think is -maybe it's the ethics education, maybe it's the professionalism education- you're developing a bad system by definition, or a failed system; one if it doesn't work, two if it's functions have negative, unintended negative impacts on other people, if you develop it in a haphazard way, not considering those impacts. That people now call computer ethics. I bring to that my value systems about

human beings. And now what I do is, you now have Don interested in professional ethics without having any vehicle or outlet for it except some of the organizations joining the ACM and joining the Data Processing Management Association and groups of that sort. That's where all of a sudden the fields began to connect. I think I've answered that question.

TM: That's your connection to software engineering and software ethics. Could I just stay with that for a moment, you said that this experience where you recognized that without thinking about, that's the problem; you hadn't been thinking about the consequences, you'd design a system that would basically be an invasion of somebody's privacy. Library records are considered to be private records. They're not part of a public record. They shouldn't be part of a police, sort of, fishing scheme in the absence of...

DG: And this particular one it was a teenager that was particularly obnoxious that the police were after and looking for anything to get after this obnoxious teenager. There was no terrorist or really justification. It was just a hassle.

TM: They were looking for anything. You said there were other examples that led you to a more general formulation that sort of predisposed you to looking towards a moralful listing even in systems, but a kind of ethical approach. Anything else from your personal life, your Harrisburg years, or you consulting years where you have particularly vivid examples either from yourself or other software colleagues that maybe you can recall? That prompted your kind of reevaluation, your broadening of the perspective.

DG: Issues, they range from strikingly minor ones to major ones. One of the issues with Blue Cross was the issue of health insurance and then abortions at the time and that period of time. We're talking the mid '70s to early '80s. Where a medical record was kept of a particular subscriber's daughter who was going to college in another state and he had all of a sudden received that he had to pay a portion of a medical bill. The girl had gotten an abortion covered by her medical insurance or a portion of the medical insurance. The information was supposed to be kept confidential, and because of the design of the system, her father found out about this particular thing that caused particular horrendous problems in their family relationship and so on, and exploded in all sorts of ugly ways with lawsuits and challenges and things of this sort. And again it was the way the system was managed, a simple kind of data protection issue.

TM: Do you find it difficult, I'm not talking about you now as a professional I'm talking about you as somebody who's interested in the craft of programming, is there a certain quality about the challenge of a small and narrowly defined problem that none the less makes it a temptation to take on a project like that? That you have to balance your interest and intrigue at solving particular narrow technical problems against the broader understanding of the consequences?

DG: Always a challenge. Always, I mean, it's part of the thing, this group of doctors, the group of administrators at a particular hospital asked me if I could develop what they call

a patient treatment form. That is, when you go in as a patient and you go in with pneumonia you're supposed to get this, that, and the other, when you go into cardiac you're supposed to get this, that, and the other. And there were some interesting ways in putting together the form that you're listening to this and going, "I can solve that. I know how to do that." And the challenge is very exciting. And then you hear them say, "And this is going to get used by us to keep medical costs down." And you feel that's a good idea and, "How will it do that?" Well, when a doctor prescribes a medication that's not on this form, the nurse will contact the doctor again to say, "It's not on the form. Do you have special reasons for this?" And if the doctor doesn't give adequate reasons then no matter what the doctor's name and the special medication will get written on a list because he's violated the standard that's written on this form. At which point you say, "So if a patient is in intensive care for cardiac problems and the doctor, if the doctor prescribes penicillin?" "Oh yes, the form will check that and that's not standard care." And you say, "But my father when he was in cardiac care got pneumonia and they gave him emergency doses of penicillin. So you mean the nurse would override the doctor's instruction until she could contact him? Won't that kill people? Do you really have all of these symptoms and medicines laid out concretely? Because you're introducing a delayed exam like it's life threatening to me." And these folks were just interested in the financial end of cutting expenses, catching the doctors who were using their professional skills to do things outside the norm when necessary. And that was one of those contracts where I said, "Thank you, no. I'm not doing it, nor should you." But part of my "nor should you" was heavy education with them and other work doing that. So yes, it was an interesting problem, and you do it or you think you're going to do it because of an interesting problem. And some of them that I've turned down, I've sat down and done just as mental gymnastics.

TM: Because of the...

DG: Just because...it's an interesting problem. It's an interesting problem painting a painting in a single color. And I will paint whole scenes in a single color just to check skills, and you get distracted. I'm getting distracted. Let me go back to your questions.

TM: In the early 1980s there were questions about not only the system, the most controversial piece of computer software that I have been aware of, or have ever was aware of, was the Strategic Defense Initiative. Did that play a role in your deliberation?

DG: Yes. In my history, I stopped doing consulting as an independent and went and worked for this bank in northeast Pennsylvania. And again it was developing system problems and so on. And then, went into teaching university but this time in computer science in northwest Pennsylvania. Now the location is important because it's an hour and a half's drive north of Pittsburgh. So Pittsburgh became the intellectual center you could go to. You knew people at Carnegie Mellon; you did some work with them in philosophy. Carnegie Mellon is a major university. Pittsburgh has a large population so people can take on radical causes and don't feel the difficulty of being viewed negatively by other people. 1983, the Strategic Defense Initiative gets announced by President Regan, and it gets very scary because one of the things you've learned is when you build

a system, one of the things you need is a clear set of requirements to build the system. And President Reagan gave as the totality of his clear system, build a shield that will defend the United States, make it impervious from the attack of the Red Menace or whatever he called it. This as a detail specification is inadequate. It got taken on as meaning a million different things by a million different people. Some saw it as [a] financial benefit to their university to get a research program, even if they didn't believe it could be done. Some who said computers aren't quite there yet to do this sort of thing. And the trust in computing despite the evidence against that trust. What was exciting and was probably a major critical element in my background was that an organization called CPSR, Computer Professionals for Software Responsibility...

TM: Social Responsibility.

DG: Social Responsibility.

TM: CPSR.

DG: ...Computer Professionals for Social Responsibility developed, and that was again a mixed bag of people. All of the raging anarchists who were anti-government and had nothing better to do but to scream and wave "we hate the government" flags. But what was more interesting was the number of computer professionals of great repute that said when speaking of SDI, "This is unacceptable. We can't do this." And I smile and go, "Yes. There are others concerned practitioners out there and I am not alone." You have the feeling, when I was developing a computer ethics course at Allegany College where I was at the time, and people were always fond of joking, "Computer ethics? Well computers aren't human so they have no ethics?" was the standard, mindless joke. And one of the things, I don't think I'm an especially brave individual so it was nice to know that there other people out there with similar kinds of views, and that I was not just a raving lunatic doing these things, saying, "When you develop systems, you need moral responsibility." So I joined the CPSR group in Pittsburgh and worked with them for a bit.

TM: And when would that have been?

DG: When I was at Allegany so it was '84... '83, '84. I'm going to have the years wrong. When I was at Allegany.

TM: Mid '80s.

DG: There you go, mid '80s, '84, '85, '86, mid '80s. And it was something that became difficult to do but I had fun teaching the students in the computer ethics course. Because what we would do is, in the computer ethics course in Pittsburgh the Jarvik heart, a computerized-mechanical heart, had just been invented and I told the students at Allegany college in my ethics class that we'd just been offered ability to work and test the Jarvik heart and do computing with that. After they got over the initial excitement the students started saying, "But we don't understand anything about that. We shouldn't test this kind off... We're not really competent to do this because we don't know this area." And then

you start relating the questions to the Strategic Defense Initiative and they would go, “Oh. We shouldn’t do that for exactly the same reasons should we.” And they found moral outrage at my suggestion that we, starting students in computing, work on a Jarvik heart that would affect an individual human life. And then when you started to look at the Strategic Defense Initiative, which had at that time putting nuclear devices in space, as the same kind of thing, even more criminal.

So yes, that had a very strong influence.

TM: Did you find networks with people within CPSR that you found common cause with that developed into colleagues? You’re professionally active in a professional field of computer ethics, but did you find people mostly through computers or additionally also through CPSR? You said that there are...

DG: The CPSR one, the period of time is important. These were at places like MIT and at Stanford and places and things like that. So I’ve done some work with some of the people at Stanford but not Joseph Weizenbaum at MIT who had, who is one of the early folks who, I think I’ve left out of this mix, also inspired me because in 1976...

[end of tape 1 side B]

TM: Okay, back on tape. Tom Misa interviewing Don Gotterbarn. We’re going to focus for the next little bit here on some of the intellectual sources that sort of framed Don’s interest in software engineering and precluded ethical problem. And he’s talked about three different people that we’re sort of pick up on; Joseph Weizenbaum, Douglas Hofstadter and Deborah Johnson. So if you want to start with Joseph Weizenbaum and...

DG: In this period of time when I was moving from philosophy into computing and starting to do things in a way where I just regarded or primarily regarded computing as problem solving and intellectual problems, I came across Weizenbaum’s book, *Computer Power and Human Reason* and I loved it because it was written by a technician and it was technically sound, but he also kept using words like “responsibility”. And it began to, as I tied it to things I was doing for various people, make it clear that what I thought were simple suggestions to business folks about how they revise their business was in fact changing the way they did business and how they interacted with people. As I was developing communication systems with Blue Cross, it had the same kinds of concepts. And that book was a sensitizer about the way I should be going.

That and the same period of time, I came across a wonderful tome by Douglas Hofstadter, “*Gödel, Escher, and Bach*.” Now this man is talking to me, who at that moment was teaching philosophy, was doing some volunteer tutoring, was developing systems as community service, and was earning a living writing computer programs. Gödel Escher, and Bach tied together all of these in one lump, and began to give me an insight as to how foolish I was to try and to keep any of these separate, and I needed to be thinking in all things at once to really develop quality systems.

TM: How did Hofstadter make that link for you? You saw that in his work. Can you say a little bit more in why you saw those three as being linked?

DG: I remember that impression from the Hofstadter book. The particular things, the issues of problem solving, the issues of incompleteness, the issues of not seeing the whole picture, I mean the Escher model of “you think you see the picture and you look at it somewhere else”. I think in fact his discussions of the Escher, what is now my favorite Escher work, is Relativity, where you have people walking around in a monastery and you look at them and they’re walking in the same direction but because of one thing it looks like the person is walking up the stairs just because of your perspective, and the other looks like is walking down the stairs. And you begin realize that if you don’t develop computer systems that take into account all of these perspectives you’re not going to have one that will work well. That in many places... it will have one that will fail in interesting ways; some disastrous, some just your system will never get used. And of course the beauty of the rhythms is wonderful, but don’t just listen to the beauty of the music because if you turn the beautiful music too loud, you’ll deafen people. So you’ve got to have the balance along with this elegance. And I think those are some of the lessons I took away and we’re now talking 30 years from those moments in my intellectual history so I can’t give you the details of the book but I know that those two books at that period were quite significant.

And I also from the philosophy background knew Deborah Johnson who was working on a book in computer ethics at the time. And she and I meet at a philosophy conference in New York, prior... I can’t put a date on it, but it was before her book was published, just before the book was published, so I guess around ’83 or ’84. So that’s the same period of time when SDI has its influences on me. As this it isn’t just my worry about silly mistakes I’ve made, it looks like, and I knew others who had made similar kinds of mistakes. But you look and you say, “The government is making that kind of mistake based on a mistaken belief about the capabilities of computer software.” Or what I believe is the mistaken belief about that.

TM: I don’t recall but I don’t think Weizenbaum talked at all about SDI or anything like that in his book.

DG: His book was written...

TM: What else do you critique of artificial intelligence?

DG: The book was written in the ‘70s. So SDI wasn’t a concept...SDI became full born March 23 of 1983 I think was when the president made his speech and sprung that on the world.

TM: These are different strings that are jelling in your mind.

DG: I mean I'm doing... It was my ethical responsibility, and my not having the language or the philosophical terms to talk about professionalism, so I use the term craftsmanship at the moment. But craftsmanship involves a not failed bit of software and not failed means more than it does the function that was described by a client or a customer. Not failed means it has no negative side effects on society and that it meets all these other standards that we've spoken about for the past couple of hours. The interesting SDI piece is that the group in Pittsburgh contained a lot of, how do we characterized this politely, it contained a lot of anarchistic, undisciplined rhetoric. And I was interested because I wanted to demand [a] disciplined approach to software development and following the commitments. So the support that came from that group was primarily the awareness that there were other people were doing it, other people interested in the same kind of thing, and my attention to these things was not crazy. That also gets added to, well at this point I'm at Allegany College and we start to get some connections with the Software Engineering Institute as it begins to arise in 1986 was I think the original seeds there...

TM: At Carnegie Mellon...

DG: At Carnegie Mellon, in Pittsburgh, which was just again south of where I was. And the connections... I left Allegany College and went to Wichita State in 1988 where there was no interest or support in the ethics issue. And my being asked at the summer of my first year Wichita State to develop software engineering ethics course material as a visiting scientist at the Software Engineering Institute was received non-positively. I had been acting chairman of the department and asked... and what it would have required was going to Pittsburgh for six weeks that summer developing material. And there was some concern about how I would manage the department at a distance and so on. This has taken me to an important point. The material, the whole event at the Software Engineering Institute in that six weeks did some positive things...

TM: You went to SEI?

DG: I went to SEI to work as a visiting scientist for six weeks in the summer developing these materials. It gave me some acquaintance with several people who were also working in the area in a variety of ways. We started this process by having a three day workshop at the Software Engineering Institute and that's where I met Keith Miller originally so that dates our relationship, our relationship dates back to 1989. And I also meet Laurie Worth from Texas who had strong interest in the same area, Joyce Currie Little, James Tomayko who plays an interesting role in this story that kind of dropped, who was in favor of ethics in project management and the ethics approach, and Tomayko I had also known from Wichita State. The positive things were coming in touch and contact with these people and meeting these people. For a variety of reasons after the ethics material was developed it was decided not to be used as a curriculum module by the Software Engineering Institute.

The work at the SEI had some, an interesting mix of consequences. The normal way of developing these curriculum models are that what's large gets developed, they get

presented at conferences, they get reviewed and distributed. So in connection with the development of this module and the SEI over the next year or so I presented this at faculty development workshops in Kansas and in Florida that were sponsored by the SEI. So I then gathered a network of people who were interested in this and understood what I was interested in. The effort and the arguments in this got some publicity and made some significant connections in that regard. The misfortune was the SEI for a variety of reasons decided not to pursue this as a curriculum module and gave the copyright of all of the materials in that project back to me. But the seed had been set in many places. And so now there was material available, being distributed, there were cases, there were approaches to how you would do computer ethics as a professional computer person. And you were not necessarily required to be a philosopher or teaching a philosophy course. And there were regular examples of how this would be relevant to the computer science person. I was not the only person doing this. My connection with Keith Miller at that time was significant, because Keith had been doing that in his own courses. So we got together and collaborated in a variety of ways.

But there is an important interlude that occurs. When I returned to Wichita, there was a political turn. And the interest in software engineering was eliminated. In that next year at Wichita, eleven of us who were interested out of seventeen of who were interested in software development left that school and went elsewhere because of the lack of support for software engineering. And I will leave those comments there. The rest of that is irrelevant. What is relevant is that I went from there to East Tennessee State University. In thinking about this whole project and all of what went on, that is probably one of the most critical elements in putting together the Software Engineering Code although it sounds not relevant at the moment.

TM: So that would be the most...

DG: Having gone to that particular university was critical in getting this thing done.

TM: East Tennessee.

DG: Going to East Tennessee State University.

TM: Can you explain why?

DG: At East Tennessee State University they already had a computer ethics course on the books, one for graduates and one for undergraduates. In its style it was a broader computer ethics course than I had been dealing with because I had focused on computer ethics for professionals. Nevertheless, no one had to be convinced that computer ethics was important. Everybody who hired me there knew of my interest in this and works in this area. They had heard me talk at various conferences. And so there was an understanding of some of my points of view and they were not contentious nor were they considered outside of computer science. The support that...

TM: This is a very supportive environment...

DG: Incredibly supportive institution and environment. And as I look back on it, had they not been this supportive this would not have happened. I can remember some of my days at Wichita where I tried to get funding to attend a conference on ethics or to present something on ethics and there was not money except if it was computer science related. Ethics was not considered by them to be computer science related. My visit to the Software Engineering Institute was totally funded by the Software Engineering Institute, not Wichita. I moved to East Tennessee State University and one of the first things I participated in was a conference in 1991 at Southern Connecticut State University which was called The National Conference on Computing and Values, where all of a sudden my network is vastly enlarged and I met, the number is wrong but I will say at least 100 people who are doing work in this area around the country. You get immediately involved in the development of a project developing the ACM Code of Ethics and Professional Practice. You find people who are willing to listen to your points of view about development, about placing items of practice into this normative code that says, "Try to write good software," and as you stick items of practice in they get put in as guidelines into this code. So the code becomes a two level code; high level normative statements, low level codes of practice. You work with Ron Anderson, a sociologist from Minnesota, who had a grant from the ACM to revise the ACM Code of Ethics and you learn a wonderful amount of things from him about the politics of getting professional societies to do things; how to avoid stepping on toes, how to move things along successfully, what elements of your discussions you make public and what ones you keep private, and the way in which you have to negotiate every level. One of the things that comes out of that is an understanding of, on my part, of what you put in public documents and how you color them in order to avoid catastrophe, avoid misunderstanding. How you just present what is relevant to the relevant parties and the work goes on where the private issues and private debates are kept private.

TM: So Ron Anderson had experience not merely in writing the code but also in having it, moving it through ACM?

DG: That's what we were engaged in doing. In that meeting, at that conference in 1990 I met Ron Anderson and worked with him on developing the code that got finished in October of 1992.

I also met Walter Maner and Terry Bynum both people who had been working in computer ethics for a long time. And I will go on; if I start to do this list it will get too long. But Gene Spafford, Eugene Spafford, and Terry Winograd also belong on that list. And Dianne Martin belongs on that list. Dianne Martin I would later work with, she would be president of the Special Interest Group in Computers and Society, Computers and Society and I was her vice president. Terry Bynum and I are now both visiting; Terry Bynum developed his own ethics center. I developed mine after his model. Walter Maner and I have worked on NSF grants related to ethics. This was quite a fertile seeding that got done at this conference for my work.

TM: This is all things that came out of this National Conference...

DG: National Conference on Computing and Values.

TM: At Southern...

DG: Southern Connecticut State University.

TM: Must have been an amazing place. Well an amazing experience to have met all these people.

DG: The event was designed to develop teaching materials so there were different topic groups like privacy issues, and security issues, and professionalism issues, and teaching ethics class issues. And each of them had workshops going on where they would produce documents, which had the grant funding supported, producing these documents and then distributing them because the issue of ethics had now moved to the forefront, and people wanted to be able to teach it. There was a very non-altruistic reason for this. It has to do with an effort by Gene Spafford who is on that list of people. The ACM had developed a curriculum for computer science accreditation. I've said this, I've got this wrong. The ACM had developed a curriculum for computer science. In that curriculum was that you must include 15 hours of social and ethical issues. That was Gene Spafford's insert I believe. That changed the world for computer science because now to be a program that was recognized in some way you had to include 15 hours, and very few people, other than those at this conference, know how to do it. And half of those at this conference came because they were going to be the assigned person who had to take care of that 15 hours of contact where you did ethics and social issues. So they were there to learn from us.

TM: So the science of the conference reflected this ACM mandate, in effect. And I'm just making sure that I understand the denomination, 15 credit hours?

DG: 15 hours of, 15 contact hours. So 3 weeks... if you need 3 hours a week, 5 weeks sometime in their career of course material would cover social issues. Could be distributed among courses, you could have a course that met for 5 weeks, 3 hours a week and you did away with it. And so that was the standard. I have to be careful, many...

TM: Excuse me Don. One hour on a semester basis is the same?

DG: Right.

TM: One hour a week for 15 weeks would be the 15 hours or...

DG: That was the Virginia Tech model for a while. But I have to be careful how I say this. Not most, not half, but a number of the people at this conference came to learn. All of us came to learn because there were all sorts of wonderful other people there. But the recording, and gathering, and preserving of the information was vital. What occurred were things like video tapes were put together; video tapes were done and then put

together after the conference. And Southern Connecticut distributed something called the Computer Ethics Starter Kit, which was a series of video tapes; some of them discussing “What is computer ethics”, some of them giving ethics issues, some of them talking about how to teach computer ethics. And so people like Keith Miller, Deborah Johnson, Don Gotterbarn, Walter Maner Dianne Martin were all on these video tapes. And if teachers didn’t know what to say they would sometimes simply show the video tape to their classes. This was a real seed. It was also a change in my visibility, I think.

TM: To appear on tape in this structured way...

DG: Well to appear on tape, to meet all these people, to connect up with writing of the ACM Code of Ethics and Professional Practice, because now I was starting to, even though I lived in Johnson City, Tennessee, a town of 50,000 people, I was now connected up with a whole network of ethics people in the United States. Some of my work was known, and I think I did okay so some of it was respected.

And the world began to change at that point. I got invited to a conference... The kinds of things that shape what goes on are amazing. I got invited to a conference at Rose Holman University, I’m going to get the date wrong, but I think it was early I think in this period of ’91 to ’92, which was called the GTE Ethics Lectures. Gene Spafford, Deborah Johnson and I were the three people who invited to give three consecutive lectures, maybe a week apart. At that lecture, before that lecture I put together an article arguing about the nature of computer ethics as being not a universal kind of issue related to every issue about computing, but that computer ethics, I argued vigorously for computer ethics being strictly about ethics of a computer professional. I took a very narrow, vigorous argument which had the anticipated effect.

The argument was... I have fun with it in my classes now because I have my students read it and I have them read it after we go over logical fallacies. And they immediately spot, shows the success of the first lecture and they’re afraid to say it, the fallacy of this because what I do is I refute a then fairly common view about computer ethics that is just for preventing you from being a bad guy. And I argue on the other extreme that computer ethics is only about helping you be a good guy, to avoid the kinds of problems I hit. And they learn that wait a minute that’s a case of the black and white fallacy, computer ethics is this range. And it’s wonderful to do in a computer ethics class because they’ve refuted the professor but they’ve also in that same moment committed themselves to the breadth of the issues.

So it’s a little bit sneaky to do but that paper was important because it took shots at someone named Donn, D-O-N-N, Parker who had said, “There is no such thing as computer ethics because computer people are all bad guys,” or something like that. It argued vigorously and gave arguments for my one side; it didn’t show that the other side wasn’t there but those arguments set a case for a domain of computer ethics that had been not strongly advocated at the time. That is, if you write a book on computer ethics you have to include a section on professionalism. And that’s been one of the things that has been, I’ve been identified with; the push for professionalism. That paper has gotten cited

places, repeated and anthologized in textbooks. And it's always fun because today when I go and meet people and we talk about computer ethics, if they have just recently reviewed that paper they gently say to me, "Don, do you still hold that view?" And you have to explain that you were responding to a position that said computer ethics was just about bad people and so the easiest way to get attention, biblical concept of hyperbole, is to go on the extreme opposite view as strong as you could. And people would look at your arguments, even if they softened them, some of them would be considered... I'm losing track.

TM: I have a specific question about Donn Parker, the fellow who you were attacking. Was this part of a counter-cultural criticism of technology?

DG: No, Donn Parker was a, is a, I think he might still be around; he was an expert in security. And he was one of the people who looked at the bad guy stuff. And he wrote actually the first sort of book on computer ethics. He wrote a book that says, "Here's this case. Here's this case. Here's this case," and he got a bunch of people together from a variety of disciplines. And he had this question, "Is this an issue of ethics, or it might be an issue of ethics, or isn't an issue of ethics, or absolutely isn't an issue," and they voted on it. And it was a description of the particular behavior of people related to computers. And then he did this first round.

TM: Of the computer...

DG: Voting. They voted on, things like, "If this computer guy stole from the bank, this computer guy cheated and defrauded money to get money for his mother's very expensive medicine," and all sorts of issues that were very social broad and so on. And then he did this first round of votes and then culled the cases to those where there was the most disagreement and did another round of votes. And then surprised in his book, "Ah, there is no agreement on what is computer ethics," and all this kind of stuff. And my argument in that paper was something about well what he was doing was picking people from, a psychologist, a philosopher, a bank manager, a computer person. And you're not going to get an agreement about the focus on a certain detail except in extreme cases. And what he was labeling as computer ethics was just these kinds of frauds.

Now that was the first book and it served a function that was a little bit positive because it got some attention but negative because it did the opinion poll type of things, like you determine truths in computer ethics by having uninformed votes. And yeah similar to the way news stations in the US now determine truth by saying, "Should somebody do this today? Call in and give us your vote and we'll let you know the conclusion."

TM: Seventy-three percent of people...

DG: And those are totally uninformed votes and no rational structure and so on. And another book had come out by Forrester and Morrison which was a collection of bad guy stuff which had it had no ways in which it'd reason in ethics, no ways in which it'd think about ethics. So this paper I wrote in '91 criticized that whole genre, and said, "Look we

can do things with reason and look there are professionals.” That gave me another level of publicity outside the department, but the department paid for that trip to go to Connecticut. It was a respectable and honorable thing to do.

TM: There was no fighting.

TM: This is back to the supportive environment.

DG: Yes, the supportive environment. It is... It is amazing. If I were still at Wichita State...

TM: A department chair that was responsible for this or who, or how did this very supportive institutional environment take shape? Do you know?

DG: I think it primarily came out of the department. Where you're limited... Deans look at are the departments doing things that bring attention and money. Departments focus on are we doing the things that departments should be doing. So we had a department chair who loved to teach programming but also had students look at codes of ethics. We had a highly respected colleague who taught the computer ethics course, graduate course and undergraduate course. We had... The computer department was a very applied computer department. We developed systems. It was less theoretically oriented. So the argument of professionalism as I've characterized it was an easy case to make with that group of people. And the dean was of course happy that the department was getting national attention.

TM: Responding to an urgent problem and getting publicity that would be envious. Did you teach computer ethics there?

DG: Not, the other person taught computer ethics for a while actually. And I'm not sure how we ended up shifting out but he, but we worked together, we'd written papers on ethics and so on. We got a grant together for the Department of Defense, DAPRA, the DAPRA agency and so on about software engineering and building things in there.

TM: What courses did you teach when you went to East Tennessee?

DG: When I went to East Tennessee I went there to set up their master's of software engineering program following the model of the Software Engineering Institute. So I taught several of the graduate courses; project management, project design, database. I taught some undergraduate programming courses. The complete collection. And this ethics stuff in the first couple of years was fit into what we call negative time at the moment.

The support was just not a question. They would go out of their way to do things. Work I did on the Code of Ethics, work I did on... Ninety percent of the cost of my work on the Software Engineering Code of Ethics was probably borne by the ETSU Computer Science Department. I got occasional snippets of money from the combined group but

when it was to go to a conference in Nashville, to go to a conference in Boston, to do those kinds of things where I was meeting with people about the Software Engineering Code of Ethics, was covered by them. To go to Boston for the final adventure with the ACM Code of Ethics that I was working on with Ron Anderson, was paid for by them.

Now that story... The ACM Code of Ethics, lots of email was done. It was done in various meetings in various places. So we had meetings in Dallas, Texas at the NECC Conference. We had meetings in Boston at a conference where we'd get together with Ron Anderson and others of us who could do it. And to do this quickly, there was a critical moment that occurred in the whole ACM Code of Ethics, where it looked like it might fail at the very last moment. There were some negative signals sent to Ron Anderson a week before the ACM council was supposed to vote. And the negative signals were sent by people in power on the ACM council that they weren't going to vote for this. We had written a normative code of ethics with some guidelines. The previous code had been a disciplinary code that said, "You shall use this method and if you don't you're out of the society. You shall use this method," and so on. And the people who like that...

TM: A method that would be a certain..

DG: Software development, or follow this procedure, and it was quite disciplinary. And they liked that because it was easy to make the decision. You follow this line or you didn't. And if you were a computer professional you look at the list and you said, "What do I do?" Normative codes give you guidelines. And sometimes there's judgment involved. So the challenge was that, "Could this be done? Could this code be used?" Anderson had to prove that the code could be used in decision making. Otherwise it probably wasn't going to pass the council meeting and the whole effort would be squashed. Ron Anderson, wonderful general, he called Deborah Johnson and I and said, "You have to come up to Boston now. We have an emergency. Bring all of the case examples you use in your classes." I told the department, "I have to go up to Boston on an ACM emergency related to the Code of Ethics." Covered and funded.

What the emergency was, was in three days in a room in Boston the three of us and Judy Perrolle, a sociologist, came in on one of the days, put together a paper, which is also published all over the place, called "Using the ACM Code of Ethics in Decision Making". We wrote that paper using our case examples, Ron Anderson sitting with a laptop, typing, and word-smithing, and fixing what Deborah and I were offering. Raising good questions and tying it all together. We printed that thing out on a Thermax heat printer the day before the conference, the council meeting. The day before, that morning we had it printed up in the business office at the Plaza Hotel, I think, in Boston, the Park Plaza Hotel in Boston. And when the ACM council came into the meeting room that morning they was sitting in front of, each person, a paper authored by Ron Anderson, Debra Johnson, Don Gotterbarn, and Judy Perrolle right next to the ACM Code of Ethics. And the title of the paper was "Using the ACM Code of Ethics in Decision Making". And our request was to approve the Code of Ethics and approve this document to be printed in

the Communications of the ACM as a guide to using the Code of Ethics in decision making. Beautiful.

TM: Can you give an example of the use in decision making? That is, you wanted it to be practical, to give guidelines not just to be disciplinary as you pointed out. Can you give us an example?

DG: One is “Don’t cause harm”. Guidelines explaining to a computer person in 1992 what causing harm is. Causing harm is destruction of other people’s files. We’re not talking about a physical harm. Causing harm is depriving them of access to their computer and so on. Given a case, Mary writes a system that would display news relevant to sailing or something like that, and the system has a fixed sized font on the display that is .8 font that lots of good vision computer nerds can read. But people with limited vision who may need sailing information could not read it. That system causes harm. Is Mary justified in not fixing that system? And you’d just look at the Code of Ethics and say, “According to Principle 1.3, don’t cause harm. Here what it says. Mary should have done this or that, or made the system flexible so you can press a button and do certain kinds of things.” So that isn’t a particular example we used. I don’t remember the article, having been written 14 years ago. But it’s that kind of thing.

It was very, “Here’s the code. Here’s the situation. Apply the code to the situation.” We know fully that that’s an over simplified version but at least it starts people out looking and seeing the relevance of the code. The point of this sidetrack that we’re on is Ron Anderson and his general ship knew what to do, and if you put paper out and you block moves ahead of time with paper and some form of publication, the moves appear silly. One of the things we’ll talk about later is some of the attempts I did with publications, which was learned from General Anderson. I guess he’d chuckle if I called him that but anyway, learned from Ron Anderson. But the other point is that I said to my department chair, “We have a critical issue with the ACM Code of Ethics. I need to go up to Boston. I need to not attend class on this day. We can cover it by such and such a faculty and I’ll do that,” and there was not a “stay here”. I can tell you if I was at Wichita State and said that, that would not have been accepted because it was not really anything to do with computing. It was doing something that I liked doing. It would be like someone saying, “Well my strawberry crop is dying,” and asking not to teach their physics class so they could go and pick strawberries. The support and I’m going to stop this point. I could probably go on. It’s simply all of the pieces were supported by the department right up to the end and all of the crazy things I did. And it would not have happened, at least my participation in it would not have been there although someone else may have taken up the ball, but I would not have been in that picture.

[end of tape 2 side A]

Development of the Software Engineering Code

TM: Just continuing on. We have Don Gotterbarn having set a foundation in philosophy and computing and software engineering, and then sort a spring board for his work in ethics at East Tennessee State. And so Don, go ahead and take the narrative from here.

DG: While at East Tennessee State University in the early '90s, I think it was '91 or '92, I attended a meeting in Baltimore of the International Conference on Software Engineering. I'm not sure what my role was at the meeting, I've forgotten that, but I know I was involved in some discussions with some people about the potential for licensing software engineers. It had started to become an issue because Fletcher Buckley, a highly respected member of the IEEE, had distributed this document. And the document became a problem because prior to the meeting he would redistribute it about every two weeks, or three weeks with significant changes and modifications. So you never quite knew what he was going to present and he had visited with various groups at the conference. One group I belonged to was the Technical Committee on Software Engineering, an IEEE-Computer Society subgroup, and he had visited with us.

And then sometime later during the conference, the Board of Governors of the IEEE met and had a meeting. Prior to the meeting there were some concerns of the Technical Committee on Software Engineering that there were confusions in Buckley's document. Some of the things Buckley had suggested were good and we had thought out an alternate strategy. I believe Elliot Chikofsky was the architect of that strategy. I went to the board meeting the IEEE, which is according to IEEE standards open to anyone who wants to be there, but I was primarily in the mode of viewer and watcher of the proceedings that were going on. Buckley presented his suggestion and Buckley, being who he was, was respected and there was no significantly negative comment in that regard but some problems were pointed out. And the suggestion at that point was that a blue ribbon committee should be appointed to investigate the suggestions in Buckley's proposal. Buckley's proposal had some standards for licensing, had some standards for body of knowledge and what he had called registration which I believe is the way engineers get listed as professional engineers. The meeting went on without any acrimony and some names were picked out as great software engineers who would be on this committee, this blue ribbon panel that would review things. This blue ribbon panel later turned into what was the IEEE representative to the IEEE/ACM Steering Committee on the Professionalization of Software Engineering. I went home, reported some of those things to an ethics list that I had established. There were some varieties of negative comments on licensing and mis-setting standards.

Then in the normal mode of events, a colleague that I had worked with at Wichita State University apprenticed the ethics elements in what I was doing and he was now teaching at Carnegie Mellon in the Software Engineering Institute, teaching their video project management course which is distributed world-wide. He would have me come down and do one video lecture on ethical issues in software project management that became part of his program.

TM: Who was the colleague?

DG: Jim Tomayko, the person I spoke about earlier at Wichita. And he also, I was also going down to make a CD. When I arrived, Jim had said something about Mario Barbacci, a name I was not familiar with but was later to find out he was a significant member of the IEEE and soon to become its president, was concerned about ethical issues in software engineering. And I immediately jumped at that and Jim said he had scheduled lunch with Mario. You have this discussion with Mario and you find out that Mario is now one of the people who was appointed to this blue ribbon committee from the meeting that had taken place in Baltimore. Mario describes to me a variety of things about the ethics standards that they are going to set to establish. I nodded and he asked if I would be interested, if I'm interested in doing that kind of thing. It wasn't an overt offer to do particular anything, just an inquiry about my interest.

I grabbed on to the opportunity and when I went back to Tennessee I wrote a three or four page letter about my stances on various things, about establishing ethical standards and standards within the IEEE. And out of that came an invitation to come to Pittsburgh and talk with Mario about this and whether I would be interested in working on the Ethics Committee. There was a meeting in Pittsburgh with Mario and a gentleman named Bob Melford, who I'd never heard of, who then chairman of the IEEE Professional Practice Committee.

I was asked if I wanted to co-chair the Ethics Committee with Melford and help in to establish ethics standards for software engineering. Needless to say, my heart jumped at the opportunity and it took great control for me not to hug these people and say, "Thank you for the opportunity." At that meeting, Melford and Barbacci who are familiar with the standards of the IEEE methods in establishing standards begin to talk. We begin to lay out a schedule. We begin to list some names of who would work on the teams with us. I listed names of people I knew like Joyce Currie Little, Keith Miller. Melford listed some names of people he knew. And the schedule was set up, and an approximate target date of November 1995 was set up for the establishment of these ethics standards following the IEEE Standards Policy. That was the beginning of my participation as ACM co-chair of this committee. In some of our discussion and in experiences I'd had with the ACM and the ACM relations to the IEEE

TM: So let me speak clear. You were there in some way representing ACM? Or as an individual?

DG: I was invited up as Don the individual and I was asked by Mario if I would be the ACM representative.

I didn't ask any of the political questions at that point. Political questions are very sensitive. I accepted the original schedule set up by Melford because I had no understanding of the policy of the IEEE. That bit of ignorance became a problem that would permeate about three years of non-progress, accepting such a schedule. And it became a political football because people looked at the date and wondered what had happened or why things had gone wrong.

Another problem in general for the early portion of the code, was I allowed myself to be in total dependence on Melford for identifying and controlling and directing the process.

TM: And this is the IEEE Standards?

DG: Yes, this is the IEEE Standards Development Process. It is a very convoluted process that normally takes five to seven years. It's very political because every company in the world wants to be the IEEE standard because that puts them in business forever. I was there talking with the president, or soon to be president, of the IEEE and a person high up on the ladder, I thought, of the IEEE structure.

We then went home and the first set of tasks was Melford's. One, he was going to write up the minutes for the meeting. We were going to contact respected people on the list to see if they would be interested in doing this. He was going to write up a document that would determine the scope of our tasks, and he would also be writing up the standard form for a call to participation that we would broadcast widely.

At various points in the early years Melford would say, "Let's have a meeting," and there would be a meeting of us at, for example, Washington D.C. at the IEEE, meeting at the IEEE, and it could be facilitated there because there was a conference going on that I would be attending and of course East Tennessee would pay for me to be there. At some of these meetings, strange things would go on that again you would have no understanding of. You had some volunteers there and in the middle of the table was a phone for a conference call, and new people who were on the upper echelons of the IEEE would be spoken to about establishing a PAR or establishing some other thing. And you had no idea; this was all IEEE jargon about the mechanics of doing things.

TM: Can you explain PAR just for us? That a particular IEEE term.

DG: That's a particular IEEE term whose meaning I've forgotten but it has to do with setting up the standard about the standard, or what the requirements will be for the standard.

And so these kinds of things went on. You'd write some minutes. You'd say, "Well we're going to get these and let's get started." We were to write up the standards of ethics for software engineering. This became a bit of a problem because when you appeared and spoke with the Blue Ribbon Committee and the minutes as written up by Melford as our meetings spoke about us as establishing something that had probably not been done before, called "Ethics Standards." When you spoke with the Blue Ribbon Committee, ethics standards were things that would help a practitioner in a particular situation describe what they ought to morally do. That is the goal was to set of documents that said, "If you're caught in a situation where someone says accept a bribe, don't for the following reasons. If you're in this situation and you need to do this technically, don't for the following reasons." And this was what most of the people on the committee had wanted. Other people would speak to you...

TM: The IEEE standards are technical standards. So in a certain way this was graphing on a process that was designed to deal with industry objectives and industry pressures onto a project that was entirely separate from setting up technical standard that commercial ramifications, and separate professional ramifications and consequences. But you were still following this overall IEEE Standards Process.

DG: Following the IEEE model. Part of it is, part of the politics of this situation gets to be very confusing. The IEEE and the ACM historically work together when they have to. The IEEE is, tends to be interested in more applied things. The ACM used to be, at that point, more interested in theoretical and mathematical things. The ACM was worried about software engineering. That is, they had set up a committee wondering about whether software engineering was even a discipline at all and whether it should be included in computer science. When they became aware that the IEEE had set up a blue ribbon committee to professionalize software engineering, the ACM did what was the right, political thing for them to do and asked to be associated and an integral part of the project. That meant that your committee, your Blue Ribbon Committee, was enlarged and included IEEE appointees and ACM appointees. One of the issues about politics of volunteer organizations and appointees to blue ribbon committees is you pick those people who have the biggest names and the most influence to be on a blue ribbon committee. But by the nature of them having the biggest names and the most influence, they're also incredibly busy and the amount of time they can devote to volunteering or working on the committee and following its detail is minimized. Some of the interesting things that that have found out years afterward, is that when you approach some of these people on the committee to ask about their work on it, they have no recollection of the existence of the committee.

TM: Beneath their notice in a certain way.

DG: Or they will deny specific things for which we have emails. The other problem is appointees to a committee are like any committee, it's loaded. And there are people who will express particular points of view. So, on this committee were ACM members who were clearly opposed to software engineering as a profession. On this committee were people who were clearly in favor of the ethics being set up as absolute standards conforming to IEEE guidelines. Against this, quietly, one or two when talking about ethics would say the word "code of ethics". At the meetings I attended of this committee it was expressly said, "We don't want another code of ethics." This confusing of requirement and what was desired added a layer of difficulty to the progress in the early years. The layer of difficulty was trying to build something that was not clearly defined, and build something that had no prototype out there that you could easily copy. That led to...

TM: May I ask? The ACM Code of Ethics that you had worked on earlier was across the ACM but not specifically focused on software engineering as a domain.

DG: That's right.

TM: So it was across the board.

DG: Teachers, mathematicians, anybody. It was across the board.

TM: So the particular problem wasn't a code of ethics but a code of ethics in this particular domain, oscillating between the IEEE and the ACM?

DG: You heard from both IEEE and ACM, "What do we need another code of ethics for?" They each had their code of ethics, why do we need another code of ethics, which was one of the issues causing trouble.

There was a second underlying issue in the politics, which had to do with the licensing of software engineers. The licensing of software engineers had at that... Part of the reason, I believe, for the acceptance of the work of this committee, or for the acceptance that this committee should work at this point, was that there was in at least one legislative bill before the state of New Jersey a discussion about whether to license software engineers. There was a discussion in several other legislatures thinking about whether they should have a similar bill. So the issue about licensing was coming forward.

TM: And Fletcher Buckley had raised that within...

DG: Fletcher Buckley had presented that as a suggestion that we should support in the IEEE. But what was happening was that there were external pressures outside of the profession saying, "Let's establish standards." The model for establishing standards that was written in New Jersey was sharply criticized by many in the software engineering professions as misdirected, misplaced. It wasn't that they were against licensing; it was that this was a terrible document. So one of the underlying motivations was, "Let's have licensing." That was an attitude that, I think, several people in the IEEE had. The ACM on the other hand was opposed to this, because if you have licensing you will create engineering school, computer science. And many of the ACM, many computer science programs were in liberal arts schools and would not be able to be, would not produce licensable people.

TM: ABET accreditation, the whole...

DG: ABET accreditation and that whole infrastructure is very volatile. So just the general underlying issue. Even today there is an ex-president of the ACM who opposes the Software Engineering Code of Ethics who thinks its existence would facilitate licensing.

Now you have this political structure where people come together sometimes through conference call and so on. You have three taskforces; one to establish ethical standards, one to establish a body of knowledge, and one to establish education, which are ex-officio members of this committee.

TM: Let me just see if I get that right. Three taskforces on ethical standards, the body of knowledge, and a separate one for education?

DG: There's a taskforce on ethical standards. There's a separate taskforce on defining the body of knowledge of software engineering, so that people won't be able to say it's immature anymore. There's another taskforce on defining the education and what a software engineering curriculum would look like.

The process that was going on was a little bit problematic in the way I had allowed myself to participate. Dependency on Bob Melford and his delivering products was a mistake. I understood the politics of ACM and IEEE and every taskforce had co-chairs to kind of check against one another. Melford I had met on several occasions would come up with suggestions and many of them he would volunteer to undertake. We would inform the committee that something that various people who were with us on these taskforces, that this would take place but things would not happen. There was a general consistent problem of communication on my part with get communication back from Melford. And then I would feel guilty and send letters to the working members of our organization at that point and try to encourage them to stay with it, something will be coming soon. Sometimes that "soon" was a matter of six months or eight months.

I've left out an important step in the organization. After the initial meeting, we had these lists of people, Miller, Little, Weisband, Patrick Sullivan and so on that we put together at this initial meeting in the Software Engineering Institute. We contacted each of them individually and asked them if they would like to work with us on this project. The names we picked were people who happened to be interested in ethics that we knew. So the positive response was accepted. In organizing this process, what we did was to have meetings where there were conferences. So in 1994, there was a conference in Boston for the National Education Computer Conference, NECC. Miller, Little, Gotterbarn, Melford, and McFarland were all in Boston. So we met at Boston College and laid out a plan. The plan in part was again addressing unclear requirements. So the attack we took was, "Let's divide the standards into sections. Let's divide them into sections that are technical and ethical." So we had a section on security. We had a section on privacy. We had a section on intellectual property. We divided the world into eight sections. The list I don't remember at the moment. At that moment in this planning session at Boston College, we took volunteers from people who were at the meeting to head what would be a working group, to write a short definition of what their working group standard would be about, and then to be in charge of running the working group.

TM: One working group for each of these...

DG: One working group for each...

TM: Security, privacy, intellectual property, and so on.

DG: Each of these was to conform separately to a standard that we would establish as an ethical standard on such and such. The one that I chaired was called "Professional

Competence”. And one of the things following IEEE procedures, each working group leader would be in charge of people who had volunteered for their working group. The meeting...

TM: This is setting up a large number of people being involved. Eight working groups, each working group having a group of volunteers. This is a large cast of characters.

DG: Right.

The minutes for this meeting were typed up and are part of the record. The idea was that then you would have a call for participation that you would set up, sent out conforming to IEEE standards. The call for participation would announce the structure for our ethics taskforce. The structure for our ethics taskforce was Melford and Gotterbarn as co-chairs and then the leaders of the eight working groups. That was the overall structure and people were invited in the call for participation to volunteer for, to participate in any or all of the working groups. That was the political super structure.

One of the things that happened was this call for participation, although not finished and approved, got out some how at various stages and different times, and some people were volunteering for things even though the structure hadn't been set up. The call for participation went out several months later. There was just a quite time when volunteers are saying, “Hey Don. What's happening? Why didn't you do this?” And I would send an email to Melford and sometimes get a response and sometimes not.

He wrote the call for participation. It went out, and we had a set of volunteers set out for the various working groups. I took the list of volunteers and distributed those volunteers by email, information on them to the various working group leaders, charging each working group to set up a mailing list, to begin work, to contact their members, to talk about how to set up a standard.

For many working group leaders there was not any follow on activity. I had tried, let's see... I tried several times to get some follow on work or to proceed to do some stuff. Sending to my working group a list of, requesting of them that they look at codes of ethics and others standards, any standards they can find, for standards of professional competence.

In the middle of this process, IIT got a grant to do some research on this process. Michael Davis sent me an email asking if he and Vivian Weil could watch what was going on in this development. I sent my group that letter and said, “Do any of you having any objections to us being part of their research?” I got no negative responses. So I added them at their current email addresses to my mailing list.

Now in this point in technology one of the things that happens is you get over confidence in the technology or you believe things are simpler than they appear. The type of mailing list at this point was something called a list serve. We had no list serve at my university but I had a friend at the University of Tennessee who got her university computing folk to set up a list serve for me with all of the emails as I had sent on to them. Now one of the

things that happened is I, as other people on my committee, believe technology works without checking it at every moment. So I would send letters to the list serve believing that people got the messages. The list serve would only accept letters from people whose email corresponded to the membership of the list serve. So when IIT changed servers, or for some other reason, Weil or Davis's addresses changed then they would not get any email about the process.

This led to one of the more interesting episodes in the development of the code. One of the people in my working group that was to establish ethical standards of professional competence was like many of the volunteers who stayed with it, wonderful and energetic and committed. Sometimes that commitment leads you to do things that aren't consistent with what goes on or leads to strange results. Edward Melcher who was a licensed engineer and working in Pittsburgh had been on the professional competence committee. They company changed his email. He received no more emails from me because his email address did not match the list's address for him. He was unaware of this but indignant that I was not emailing him. And he had a list of the people, other people's emails from the list serve from when it did connect with him. So he sent to several of them, about six of them, saying "Is the Professional Competence group still in existence? I haven't heard from Don Gotterbarn in so many months." And they responded back some of them having heard from me, some of them not. All of them choosing to blame the human being for impoliteness or whatever because he was not keeping in contact with them.

The episodes with Melcher who was a significant contributor to this whole process, another one of those people who I would probably say this would not have gotten done had it not been for him, is that he changed jobs a few times, changed emails a few times, and contact with him throughout the whole process was always amazingly difficult. Contact worked through his sending to his own list of a small group of people. At this time mind you, you had eight working groups, six of them having done nothing. Six of them, not being inspired very much because of delays in learning the scope and procedures of standards and so on and having their operations not clearly defined.

There was during this an event that occurred in, '95 national meeting... There were some higher level steering committee political difficulties that we as ex-officio members had no idea of things going on. Some of these items had to do with the other two taskforces. In '95 for example, I was at a meeting in Nashville where the Steering Committee came to have its meeting, but ex-officio were excluded. Only paper reports from the committees, from the three committees were allowed. And they discussed issues with the other items and you knew there was some difficulty because you weren't even included in the discussion. Although you met with Mario Barbacci who was now, I think at that point, heading this group. And Mario in private conversation said he wanted the ethics to succeed and was very strongly in favor of it. So you knew you had some political strength. The ACM member, Dennis Frailey, was a primarily support of the ethics process through out the whole thing. In fact most of my communications to the Steering Committee were not responded to unless they were responded to by Dennis Frailey. Melcher...

TM: Just a second, if I could ask you to reflect on this at the moment. It is the difficulties that are going on you might say that it was adopting the wrong model, an IEEE Standards model, for something that was different. Alternately you could say that it was the wrong type of people, or the wrong type of number of people regardless of the structure for, I'm just sort of curious, the six out of eight working groups didn't sort of catalyze. They weren't energetic. They didn't do the things that they hoped to. Was that a failure of individual leaders just to take up and to organize those six different working groups or do you have a diagnosis about sort of how this, sort of, inactive time went?

DG: I have a diagnosis about the inactive time but I'm about to respond in a way that is just like the response that you worry about when you make documents public. We had, in Pittsburgh at that first meeting, named the people we wanted to talk to. Some people when you ask them to do something will say yes and they get their names in lots of places. But they are distracted by other things or there are people in the world who dream wonderful dreams and don't know how to follow through on them. We have the people who were interested in the subject but were not what I call "closure people". That is the follow through on tough stuff leads to, or is caused by numerous distractions. I know that one of them their spouse got sick. Other ones you just never heard from again. But now you have this publicly identified group, some who were IEEE, some who were ACM representatives, and to do anything to upset the balance are the kinds of things that are quite difficult.

***ACM Meeting of 1996**

Let me give you an example of one of the items that went on. Early as our committee was established as Melford was to be writing the various IEEE standard boiler plate things, we attended a meeting, or I attended a meeting in Pittsburgh which was open to the ex-officio members of the Steering Committee. And gave a creative progress report that we had decided on eight groups and done this kind of thing, and did not say a whole lot about the non-progress of the groups and so on.

Part of the thing that was going on at that meeting was one of Melford's acquaintances in Washington was setting up a special computer issues meeting with the House of Representatives. Where we were, they were going to talk about and present to the House of Representative the various kinds of ethical issues and issues with computing. And I think the person's name was Peter Tipton or Tripit who was responsible for organizing this, and he was the one who contacted Melford. Melford told me and told the Steering Committee that we had been requested to do a congressional briefing on the development of ethical standards for software engineering. A press release was written for that, drafted. At the meeting I attended the press release was presented to the Steering Committee. Now you're talking about people who are on the Steering Committee who are the blue ribbon folks, who are the top. Well the memo started, "The IEEE Computer Society, 100,000 members strong and the ACM 85,000 members strong have joined together to establish the professionalization of software engineering," baboom, baboom, baboom. "And are going to do this congressional briefing." It took ten minutes to get past

the first sentence because the ACM people did not like the fact that their membership was listed at 85,000 whereas the [IEEE] membership was listed at 100,000. The ACM representatives wanted the memo to say, "The joint committee, the ACM and IEEE, representing 185,000 computer folks," covering the discrepancy, and the memo, the analysis of the memo went on like that. This is just a simple example to give you the idea of...

TM: The prickly nature of the...

DG: Yes, very, very sensitive.

TM: Just by the way you've been saying IEEE, in 95% of the cases you don't mean the IEEE at large but the IEEE CS.

DG: Yes, that's correct.

TM: That's fine. IEEE's fine.

DG: This was one of the political nightmares because you knew of the most recent debate, which had been which of the two was the older computer society. And the IEEE had come up with some evidence, which they believed made them one month older than the ACM and so they were advocating this. And the ACM then in response simply changed the name of the ACM to ACM: The First Society in Computing, "first" meaning premier. That gives you...you have to know...

TM: So there's a lot of tensions and turf...

DG: And you have to sit there and watch all of this and every move has to both move forward and not upset enough of the opposition so that they raise issues to stop the forward motion.

TM: This is we would say contested terrain.

DG: Yes. So we've done the Nashville meeting... Ah! Then comes... we're up to Maltcher. Maltcher having done... Melcher having started gathering the stuff together, Ed sent me an email and said, "Would you mind I have this sub group ready to do things and if I led them in gathering this stuff?" And I was pleased that anyone was working so of course I said yes and encouraged him. Then there was silence. I presumed Melcher changed jobs again. Communications with Melcher ceased.

On a totally separate thread, Melcher's group had put together, as I had requested initially and Melcher encouraged them to do the same thing, a list of ethical kinds of approaches to things; requirements, imperatives. And Melcher had to his group suggested a preliminary draft of this, of how this would be put together as a code of ethics. Now this is one of the other interesting left turns that the process takes. The initial tasking was standard, standard, standards on a code of ethics. From what we've said before, you know

about me that when you say standards to me I tie that intimately with ethics. So underlying a heart-of-hearts is, "Yes, if we can get a code of ethics out of this we're really going to win."

[end of tape 2 side B]

Okay, so Melcher is doing a job of getting this material together and gathering information as was requested and as he produced the same tactics and is trying to get a structure to it. At the same time there is little response in the needed items from Melford and now we're turning a corner and we're into 1996 from 1994. In February of 1996 in Philadelphia, the Association for Computing Machinery had one of their standard meetings. So we called a meeting of our ethics taskforce. At the ethics taskforce meeting, Miller, Joyce Currie Little, myself, and Bob Melford appeared. Interesting things went on at that meeting. The first thing was Bob Melford did not come to our taskforce meeting but saw me and was mumbling funny things to me about he didn't know if he could be involved anymore. Without any kind of explanation, very cryptic statements and you thought he was mentioning some things which indicated the IEEE might have some disapproval of his action. We'll leave it there.

Also at the meeting Dennis Frailey came up to me before our taskforce meeting and said the ACM is considering dropping their support. I had at the moment read him as giving me a personal message saying they were not going to support the ethics project any longer. He said I needed to produce some deliverable, like a schedule and a plan, and it had to be given to him to present to the ACM committee. Frailey was continuing his role as inside supporter of the ethics project as ACM representative. Later, I now believe, that it was more significant than I had interpreted it as being, because the ACM, a few years later, totally backed out of its support of the project on all grounds and no longer participated in it.

Frailey's message to me was that I had to put together a schedule. At the meeting of the taskforce in Philadelphia we then put together a schedule. Again, overly optimistic. Let's get the taskforce leaders working. Let's get the information input from them. And Keith Miller volunteered to use a portion of his summer to write out the standards documents that we would produce and put them into a consistent and eloquent form. Because he was giving up his summer, which would normally involve some income, we were going to ask the Steering Committee for financial support for the scribe. The meeting ended with an agenda, the schedule set up, and my task to request information and funding from the Steering Committee.

The request went to the Steering Committee. The request went to all of the working groups. The results were no response from the Steering Committee so no financial support for Miller. Miller's group and the Professional Competence group were the only two that ended up doing anything. Going through the year, those people who were not in Melcher's group would respond to me saying, "What is going on?" And we couldn't get the work forwarded because I had no approval from my co-chair who was now totally out of contact. I sent several messages to the co-chair; his office, his home, left voice mails

listing the things that had failed so far and not been done. And told him I was going to go ahead and work with my Professional Competence group if he didn't have any objections.

This is a strategy I strongly recommend. To get things done, you don't wait for approval. You work out what you think is right. You do it. And when someone comes back and says, "Shame on you," you go, "Oops." But you've gotten the job done. One of the ways of doing it is to consider non-response as approval. With Melford, although the structure of the committee was I needed response from the, support from the IEEE side before I could do anything as a co-chair, I simply said, "If you do not respond, I'm going to do it." I had reached a significant level of irritation at being the goat, the person who said we're going to do x, y, and z and then nothing happened because I didn't have significant support.

TM: But you now took Melford's lack of communication as a lack of obstruction and moved forward?

DG: No, I forced that position. I said, "I'm going ahead," and given the fact that he had not responded to anything I could reasonably expect that he was not going to respond to this. I have still to this day not a clear understanding of what went on and so forth. I know from discussions with one of the presidents of the IEEE that there was some problem with his leading the professional practice organization but I don't know what those problems were.

So I started to go forward. And again progress was slow. We're in the middle to the end of 1996 when I'm worried about going on a sabbatical. Six months of which are going to be spent in Washington D.C. and the next six months spent in England. While this is going on, [Mechler] has contacted Michael Davis who was doing this research on the project, and convinced Michael Davis to draft these documents into a code. Again this is one of those interesting notes that a code is not what was asked for but a set of standards was asked for. Davis wrote the code and I believe it circled a few times in the Mechler group. Mechler sent the code to the new head of the IEEE-ACM Steering Committee.

TM: I'm sorry. Did I have Melford asking Michael or Mechler asking Michel?

DG: Mechler, I'm sorry. This is going to be a problem. I'm going to transfer the word to Ed. So we'll talk about Bob as Melford and Ed as Mechler. So Ed asked Michael Davis to write the code. Michael did write the code and it got circulated into the sub group of the professional competence sub group led by Ed. Ed, thinking I was out of the loop and not corresponding with him and unwilling to do so, I having gotten no emails from him because he couldn't respond to the professional competence group, sent the code as if it was a final document to the then named head of the Steering Committee as the contribution from the ethics taskforce. Philippe Cabrerra had been appointed head of the Steering Committee. Philippe Cabrerra was working for Microsoft on an absolutely major project, which took all of his time. Philippe had little understanding of the politics inside of the Steering Committee and when sent things by Ed, Cabrerra responded as a gracious

leader and said things like, “Thank you for working. We appreciate it,” and so on. Philippe then circulated this draft to the members of the Steering Committee. Given all of the structure that I said before, you can image the difficulty that that created. Those that didn’t want a code of ethics objected. Those who feared standardization objected. Both sides objected in a variety of ways.

And I received an... And at this meeting one other event occurred, probably the most significant from my perspective. Bob Melford who travels a lot to conferences was in Pittsburgh at a conference. Bob heard that the Steering Committee was meeting and no ex-officio members invited. It was a closed meeting. Philippe Cabrerra didn’t understand the reasoning for this. Melford appeared at the meeting and sat down, and Philippe invited him in conversations.

It was reported to me afterward that Melford declared that he was totally aware of Mechler’s code of ethics; again I must repeat that this was a thing done among a closed group Davis, Ed, and a group of his people, and Bob having been out of contact for almost a year had no idea about what was going on. At this point they gave a series of charges about the code of ethics to Bob. Bob said yes he would get it done and took some gracious thanks for his efforts on the code of ethics and things he did.

I got a call from Dennis Frailey while I was now living in Leesburg, Virginia on my Washington sabbatical and I can remember that call and where I was quite vividly. It occurred in the evening while I was down in the basement working on my computer and the phone rang and it was Dennis, and he asked me, “What’s going on with this code?” And I said, “What do you mean?” And he told me about Melford accepting that he would do x, y, and z and Don and Bob would take care of it, and I exploded. I said, “I’ve lived for three years like this and I will not do it any more. I’m about to resign this committee. Nothing has been done and I can’t have my name dragged about like this.” I will confess to being an egotist. I don’t like my name in the mud. And being accused of not working and being responsible for all of the slow down offended me. Now I hear that, I’m going to say this carefully. I was upset by the description. It was reported to me that Cabrerra let Melford speak because he didn’t know Bob’s previous history and what was going on. That Cabrerra was still primarily a busy person who had very little time to do this.

That evening I composed a letter to Frailey and Cabrerra saying, “I’ve delivered projects before. I’ve delivered them on time. This project has been at an absolute stand still because you have a monster with two heads. Monsters with two heads starve to death. I will pull out of this or I will take over. If I take over you will get a product. What would you like to do?”

I received information back that I was to be put on as the head of the committee and to do whatever I thought necessary to make things work. A formal email got written by Cabrerra stating that and in a general way stating that for a variety of reasons people are unable to work on a project, Gotterbarn will be in charge. I found out years later when Davis was writing this book that apparently Melford resigned. I’m not sure when that happened, whether he resigned right after he said he would do all these tasks or how that

worked. I have no idea of the politics. What I know is that I was put in charge. And I though, "Now we can go ahead and do things," and I proceed with my side of the story. The first thing I did while in Washington was to restructure what had been going on but very concerned about the politics. So what I did was to take my professional competence mailing list, which was a handy tool.

TM: And you're restructuring not your working group but the whole committee structure?

DG: I am now the lead of the taskforce. That is correct. The restructuring is done politically so that I can eliminate or throw off heads of working groups that may not have been doing any thing. I have a list called Professional Competence that has nothing to do with that working group anymore; it's going to be a tool I will use to structure everything that follows. I send to the leaders of the working group an email that says I'm put in charge. I tell them that they will receive letters from the list serve in Tennessee. Now remember I'm now in Leesburg, Virginia trying to get a Tennessee list serve fixed that's being done sort of surreptitiously at the University of Tennessee because they're doing it...

TM: Not your home university.

DG: Right, I mean this is... This complication continues through. So people think... Of course the Davis, Weil emails are changing and they're wondering why they're getting stuff or not getting stuff. What then happens is to keep things politically correct I email each leader of the working group. All but Miller's had done nothing. I'm sorry, the professional competence group and Miller's group are the only groups that had worked. And say to the leaders, "I'm adding you to this list. Please take the role of communicating with any of your volunteers that have worked." My belief at that moment is that none of them have ever contacted their volunteers, but I need to keep this people on board to keep the IEEE/ACM balance for political reasons. So that's the politics of the situation.

I also know when I am going to England within a few weeks to meet with Simon Rogerson who I had met on other occasions and who has the characteristics that are necessary to make a project, a volunteer project complete.

Is that going out of ink? Oh, okay.

The characteristics are that when you say you will do something, one you mean it, you take honor in that claim, and you work in negative time if necessary to get it done. So the primary move I then made was to take what I had evidence of what my working group had done at that moment, which was the people in Mechler's group, in Ed's group, Keith Miller and whoever he brought along, and Simon Rogerson and know that these people were going to be the workers on the named taskforce. And there's a difference between the named taskforce and the worker bees. Then what I did even further...

TM: And the named taskforce above all else means needs to preserve this IEEE/ACM balance.

DG: This was my belief.

TM: Nonetheless you also need to get work done so you need to find people who are capable and active with that. And you can to some extent not worry so much...

DG: I can complete a project if I pick the right people. The people who want to work and are committed will do that job, and many of them who were on Ed's original group who carried through to the end. There were some people who appeared now and then out of the woodwork who were amazing, who did wonderful things in the project. And you keep saying, "How is this happening?" and it continued to go on.

So what I did was to use a name to legitimize closer control over the project. So I invented something called an "executive committee." And to legitimize that politically I wrote a story, I've got to say this better. To make this politically correct as to the makeup of the executive committee, the executive committee being Gotterbarn, Miller, and Rogerson, I clearly labeled them as Rogerson as a member of the ACM, Miller as a member of the IEEE Computer Society, Gotterbarn as a member of both societies. We represent the societies then equally. Miller is a practicing, was a practicing software engineer and he also not only a college professor, but he's built systems in the US. The international prospective that's desired is represented by Rogerson. So this is what I sent in as my structure for the committee.

Then came a surprise that is, depending on your interpretation, indicates a belief that I could get things done or were hurdles thrown in front of me. What was asked to be delivered by February 12 were three documents; one being the an explanation of the architecture of the code, one being an explanation of the functions of a codes of ethics, and the other being a justification that this was an engineering code and what were the specific engineering elements. Three documents, I'm arriving in England, and in less than two or three weeks these things had to be delivered as final documents. All sorts of things take place now.

TM: Just a second. Those, the three documents on architecture, functions, and justification are being delivered to?

DG: The Steering Committee.

TM: Steering Committee.

DG: One of the presumptions is that they'd all actually read them. There was a negative tone in there I hope you detected. You were doing this, I believed, as an exercise. So part of my goal was to fulfill the exercise. Davis, in his book about this project, complained that these documents did not represent extended doctoral scholarship about this, but that was clearly a charge based on a mistake that function that was intended.

At this point all sorts of wonderful things go on, but all sorts of things that are indicative of problems that we had originally and continued. In the printing of the table I had, people who were on my Professional Competence Committee contributed, Mechler's committee contributed, Miller contributed, Rogerson contributed. We had a PH.D. student who was working at the center there with us. We put him on grunt work looking at code and putting things together. The table got developed and distributed to the committee.

I'm sorry, I've got to say this clearly now. I have a policy. I tend to hold opinions very strongly and assert them. Sometimes I can be dead wrong, knowing that I can make mistakes, I always take the safe road and say, "What do you think? What do you think? What do you think?" And send it out to whomever I trust. I now had a group; Mechler's group, the executive committee, what was left of Miller's group that I could trust. So this thing went out to these groups. And of course the email nightmares begin, because Mechler was unable to print this particular document. And a back and forth exchange started. And it was a little bit embarrassing because Mechler didn't understand a basic thing about the emails and about the kind of font you would print the table in. So he had used relative font which changes sizes depending on which letter you use. So the table never came out right. And we were all, all of us on the committee now wondering, I'll have to speak carefully, on the ethics taskforce trying to speculate as to what the problem was, because we did not understand that he did not know this basic truth.

TM: To do a code is to do it in Courier. So it's standard.

DG: Right, you set it in Courier.

TM: So a "w" takes up the same width so all the columns stay lined up.

DG: You know this basic truth but this one great person who contributed to the code didn't. Although he's working as a software developer in various companies in Pennsylvania. And we spent days fooling around until one of them said, "Maybe Ed doesn't know this." And we looked at each other going, "You've got to be kidding." And it got sent out. Now those are silly kinds of things that go on but now the process of the code is underway.

One of the other things I sent out was a schedule to the steering committee using the very same method I used with Melford, with Bob.

"Dear Steering Committee,

Here is a very tight schedule. We'll finish things by December. Here are the things we want to do. Here's what we need from you. If you delay, you will delay the project. We need a budget to put up a web page to exchange things. We need to," now the publicity standard comes in, "if I do things publicly you can't say I'm not doing things and it's tough to deny. We want to print the article. Here are the versions of the code here, there, and else where."

And you lay out the schedule that has the versions of the code going through various test phases and so on. You sit on this for a little bit because you're not hearing from the Steering Committee and you can't let out the schedule publicly because you don't have funding from the Steering Committee, and one of your elements involved funding. Consistent with what went on before, no response, no funding, but I smiled because the letter had said, "This is what we're going to do. If you don't like it, object." And there was no objection, which meant I took it as license to move forward, and that's what I did.

So the code of ethics then started to go into version 2, and you started to encounter some of the interesting things with English English and American English.

TM: Just checking. You're now doing this work...?

DG: I'm sitting in England.

TM: In England.

DG: I'm sitting in England at Simon Rogerson's university. In a very small office this one element becomes a really interesting positive. Which was really an ugly place; it was what had been a storage area next to a staircase. Had no communications with the outside world, most people didn't even know I was there. This worked out to be wonderful because Rogerson would come into my office, we'd shut the door, and neither he nor I would have distractions. We'd get comments from people... We had enlarged the taskforce by inviting some people in England and some people in his university to participate in this as well. One of the things... And Keith Miller in the US is one of those people who he'd get an email and you'd get a response back in two hours. That was the way he operates always and to this day he does. And so we would get suggestions coming back and Rogerson would come into my office. And we would start to go through the suggestions and we would start to put this thing together in a list in so on, send it off to Miller, "What do you think? We think this. What do you think that? We think this." And it would come back. One of the early decisions that was important was this code is going to be in American English.

TM: Not in British English.

DG: Not in British English, and then you have spellings with "honor" and "favor" and all this kind of stuff and you just avoided it. And that became a model for "The Code". Everybody was invited...

TM: May I ask, were there significant intellectual inputs? In other words, did certain aspects of the code change either their intentions, or aspirations, or consequences from you're being in England at the time? You have people from Rogerson's university working in your group but can you reflect a bit and say, "Well the code is different because we have British people working along side what was otherwise an American group."

DG: No, and what is interesting about the ultimate progress of the code is that a certain theory I have seems to have some support. My theory is that a software professional no matter where they are in the world tests their software, writes good documentation, makes it maintainable, builds it the best they can. This code... Mechler's committee had an Egyptian on it. My final committee, the final taskforce had El-Kaldi, the Egyptian on it. We had people from Australia on it. We had people from all over the world.

When the code was finished, I received a copy of it from Croatia where people had translated it into Croatian and published it on their website for software developers in Croatia to follow. Shortly there after, I received one from Shanghai where the Shanghai Computer Society had adopted this code for them to follow and it was a Chinese translation of it. I received three requests from Spanish speaking countries, Spain, Argentina, and Mexico, for a Spanish translation. And I did my typical thing of, "Okay, you guys want to do it, I give you the honor of writing the first draft of the Spanish translation." And we have a Spanish translation.

And you start to get this and you go, "The code..." People have argued who tried to do an international code dictatorially by one nation writing it and spreading it failed. And they write all sorts of stories about why you can never have an internationally adopted code. The Australian Computer Society has just adopted the Software Engineering Code and made it a standard of practice for the engineers in Australia, not just the Computer Society members. And it goes on.

To point out particular ones, there were styles of doing things that were not particular items in code. The Europeans took more of what's called a virtue ethics approach. This is, lay it out and trust on the person to be a good person, to here's the standards and you follow them as best you can. The Americans took a rights obligation; state it as a fact, mark it down type of approach. That was the discernable difference from nationalities. In [the] whole process of doing this there was little about issues with, that you could identify with country-wise. The use of the word "assure" and "insure" caused some, I'll say some silly grammatical discussions back and forth.

I have to go back one step when I talked about Bob sending the thing to the steering... I'm sorry. I have to go back one step when I talked about Ed sending the first draft of the code to Philippe Cabrerra and so on. One of the things that happened...

TM: You're returning to your narrative.

DG: Yes, I'm returning to my narrative of that time. I believe it almost killed the code. The reaction from the steering committee is, "Unacceptable, unacceptable, unacceptable. Justify this." Now they only had two people writing out comments but this was the representation from the Steering Committee. So some people... It's assumed that only two people on the Steering Committee objected to what was going on because only two people wrote about it. That was a mistake. There was in the conversations with me on the phone it was, "This is going to die. There is this problem." And that was my "Well I'm willing to take it on if I can have the responsibility. You can't use my name on this any

longer unless you give me the responsibility.” And that was that transition but that it almost absolutely killed the code.

And I tend to think some of the things that went on later were kinds of hurdles that were still put in the way by those that were opposed to the code. Because the only thing you heard after we got version 2, or a revision of the first version done... I wanted to put more pressure on the Steering Committee to get some support for a website. So I sent the follow up message that said, “Where are you guys? Haven’t heard from you.” And that was the tone of it.

TM: To the Steering Committee?

DG: To the Steering Committee. “Where are you guys? Haven’t heard from you.” There was no question; we now had a group that could rightly be called a working group because they all were. It was great fun. Now Davis pulled out of that and I didn’t hear anything more and didn’t even know about Davis’s original contribution to the code until almost six months later. And I sent to the Steering Committee, “Now we’ve got this deadline for this publication,” and so on. And they didn’t respond and so we got it into the Bulletin of Computers in Society, and got an edition of it out.

After that edition hit the press we got a strange message from the steering committee that what they wanted was for us to say in the front of any future publications to have a, to write a memo that says that this isn’t really the Code of Ethics. That it is out there for consideration and it doesn’t have any lawful standard or anything like that. And it was a “minimize the publication” and they had appointed someone named Stewart Feldman to write this memo.

Tactics come into play there. You knew that if this memo was written in the legalities that would said, no one would read the code with any impact. So what we did is another standard tactic that you should do on a committee. That is if you’re worried about the way they go, you take control and do the work for them. And then rather than them doing the work, they’ll accept the work you did. I don’t know if I said that clearly.

TM: Yeah. Instead of having this outside person draft this sort of framing memo, that you would draft the framing memo in language that allowed that this wasn’t an official document but won’t water it...

DG: But won’t water it down. But I mean that’s a standard approach I use that when I’m worried about the way a committee will go, I will volunteer. And I’ll write the minutes, or I’ll write the whatever. And that will make sure that some of the things I worry about, maybe mistakenly will not occur. So we wrote the memo based on some things that Mario Barbacci had written for the Steering Committee a year or so earlier. So Feldman accepted this and we put it in the front of future publications.

TM: So the version went out in the Bulletin of Computers in Society had no cover. It was just stated.

DG: Well it was stated that it was a draft and that we were looking for comments and so on. I write it by that because that was truly its intent. You know, where are we going? Interestingly enough, when you ask Stewart Feldman about his involvement recently in some work being done on the Code of Ethics, an attempt to interview him has been denied because he's said he's had nothing to do with this. But emails to and from him are in the archive related to this.

So this then proceeds and we now start to get comments on version 2 and continue this iterative process. And documenting everything we're doing all along the way.

TM: I do have one thing, can I ask? The version you were sending out, you consider that to be version 2? The one that you were publicizing, putting out in the Bulletin and all that?

DG: Right, version 1 that was written by Michael Davis didn't make any publication that I know of other than things that Michael has written about it.

TM: What were you looking for in the comments? In other words, were you looking for all kinds of substantive ideas or was this more of a question of trying to get in touch with people who objections and might try to push the code one way or the other? Because when you asked for comments, you could get thousands of comments and the job of sorting through them could become out of control.

DG: Yes.

TM: You said that with a smile by the way, a knowing smile.

DG: Yes, you know you invite thousands of comments but you've just worked through with eight working groups of which you've gotten material from seriously interested folk who either hate it or love it or are committed to the project. So the comments you got were things like, "You know this really doesn't belong here and we should have a management section." "You know this section that describes discrimination, misses out on this." Comments, hoping for better completeness, better articulation. Criticisms catch text that you write that could be subject to misinterpretation. So if someone misinterprets, if someone reads this in a certain way, I have the assumption that it isn't their fault that they're stupid, it's my fault that I didn't write it in a way that they could understand.

TM: It's a bit of user testing in a sort of way.

DG: Yes. And correcting it so... And the presumption being that there's some problem with this. Now admittedly we'd sometimes get comments, people at Microsoft said it's not the developer's responsibility to do testing; it's the customer's responsibility to do testing. And we flat out rejected that.

TM: Curious.

DG: Two people from, only two people from Microsoft commented and that was the same comment from both of them, which was strange. And that's the kind of thing that you say, "Well that doesn't fit what we understand software engineering to be." You also were looking to capture what the profession thought software engineering to be, not what Joe of the ACM, and Heinz of Germany and so on thought but what the profession thought.

TM: A kind of reasonable center of mainstream opinion.

DG: Now some of the things would in some context appear controversial. Like you have a duty to report when someone is doing something wrong as a software engineer and you know it. Even though it isn't in your project, if it in any way damaging to society or the profession you have a duty to talk about it.

Other things we did while building the code was to address known criticisms of codes of ethics. There's a standard article by a philosopher John Ladd written twenty-some-odd years ago, where he criticizes codes of ethics because they're self-serving to defend a profession against attacks, that they don't give you any way to use the code, that they don't help you prioritize issues. Like the Ten Commandments have ten absolute laws and if two of them occur in a situation in conflict you have no idea what to do. So one of the things we did to address this is was to build a preamble to the code and to add significantly to the front end of the code, not the imperatives, but a statement that says, "This is what this code of ethics is intended to do. When you are in situations that have conflict, do this. When you need to make your judgments, this is a way in which we suggest you order things."

In this version of the code, we made, I made a mistake, two mistakes. One mistake was...

TM: Maybe we should hold there and continue the two mistakes on the next side of the tape just so we don't cut them off.

[end of tape 3 side A]

DG: So as we're moving along, following this schedule that I'd laid out part of the schedule had asked for the Steering Committee to define for us very early on what the formal process of approval would be once we had written a finished code and gone by all the standards I had outlined. That again, never came back from the Steering Committee. And it was again by design on my part and others' suggestions that we move this thing through in a variety of ways. But part of the process was to get public review at every stage possible. We had now version 2 that was ready for revision and we were getting comments from people and voted on by the ethics taskforce, the group that was now working on the code, which had a significant number of people. And version 3, while version 3 was getting ready one of the mistakes we made was to have the graduate student at the center write out the ballot that was going to get published with it. Our ballot was one of these five part "strongly agree, agree, no opinion, disagree, strongly disagree" type of ballot and the graduate student was just to list the clauses and put summary

statements. And the mistake might have been better characterized as us not carefully reviewing that ballot but thinking it was okay because it fit our first glance. The problem was that the way the ballot was stated for I think it was Principle 2.5 at that point, which said, "Respect cultural diversity." Now when you get into the United States and you talk about cultural diversity and respecting cultural diversity it is a very political hot button and it does not adequately describe a clause that is intended to say, "Don't discriminate." So that phrasing we think lead to a negative response. That was one mistake.

The second mistake I think in version 3 was my mine. I had a number of years before written an article trying to explain what I considered to be the levels of computer ethics. Characterizing one level as the level of standard humanity; we don't hurt other human beings, we help human beings and so on, and we act respectfully towards them. The second level being the level that professionals have in that all professionals owe a higher order of care. The third level being the particular profession's standards. So owing a higher order of care as a software engineer means I will design a human-computer interface that won't hurt you. Owing a higher order of care as a psychological counselor means that I will not reveal private information about you and so on. So I had this wonderful theory. Now one of the criticisms early on of the code was that version 1 had some elements that looked aspirational and some elements that looked disciplinary. And in that, those early criticisms of version 1...

TM: Just to fill in the record again I think we talked about disciplinary before but could you make the contrast to aspirational and disciplinary.

DG: Aspirational doesn't tell you a specific idea, it says try to do the best you can. Disciplinary say do this specific thing. Although version 1 didn't have specific penalties if you didn't do the specific thing, the Steering Committee...

TM: The earlier ACM code had been heavily disciplinary.

DG: Right, and then it got replaced by a more normative one. The Steering Committee in their comments, in their published comments about version 1 were worried about the distinction about aspirational and disciplinary being in the same code. So what I thought what I would do is in the preamble stick a distinction that I just mentioned to you about levels of professionalisms. So that in some places a code of ethics can simply say, "Be honest and don't hurt people," well that's one of the levels of professionalism. The other place it can say, "In your work as a software tester do this kind of testing," and that's the third level of professionalism I characterized. So I put this in the preamble and version 3 is now ready to go out to the world and I'm ready to head home to Tennessee from my visit in England.

According to my schedule it was supposed to be published in the IEEE Computer, which is the main, the flagship journal of the IEEE Computer Society. And it was supposed to be published in the Communications in the ACM. Following a model that had been, when asked about, actually supported by Mario Barbacci to me in our private conversation, I

sent the journal, I sent the article to the Communications in the ACM and to IEEE Computer.

TM: Are these transactions or communications?

DG: Communications in the ACM. CACM is the abbreviation. And called each of them saying, "Here is the article as agreed upon in my memo to the Steering Committee represented by," and it was then one of the people who would become president of the ACM and then one of the people who would become president of the IEEE, "And this is our agreement to publish the process of this." I've already laid out the approval process that I spoke about. That is they had no comment so that was approval. The publishers went and agreed to publication. It got a little bit...

TM: In both...

DG: In both journals. One of them, the IEEE Computer wanted to publish it in October. The Communications of the ACM wanted to publish it in November because they had this long lead time for articles of that length. We wrote a wrapper on the front that said... We wrote an introduction to the code that says here's what's going on, here's what the committee has done, here's the code, here's the ballot, please fax the ballot, and send it in. We had no support for a website so you couldn't, we didn't say we have no support for a website... This thing was to be commented on and sent it. Both journals agreed to publish it. ACM in November, IEEE in October. IEEE publication goes on and you hear nothing about the Code. Then you look at the IEEE Computer and you realize that the article is published in the middle of the journal without a reference to it appearing in the table of contents or anyplace else. So unless people read the journal from cover to cover they would not know that there was an article in there.

TM: It was not listed on the title page.

DG: It was not listed in the table of contents.

TM: Table of contents.

DG: Or any place else on the document. I called the editors of IEEE and I called Mario Barbacci. Having the right and powerful people on your side is also a significant way to get things done. Mario was on the side of "let's get things moving". And what happened was they agreed to publish it again the following issue; this time in the table of contents, this time an announcement on the spine that it was the draft Code of Ethics. That becomes the November event. Then we have all of the ballots coming back.

TM: And your belief is that Mario asked the editors to do the...

DG: I've forgotten which way it went. It was either Mario said, "Yes, use my name," or Mario made the call. I don't remember the way it went. It was with his blessing, and you have to think of these editors who have consumed four pages in one edition and four

pages in the other edition and say exactly the same thing. And there's a certain ethical issue here because somebody's article probably didn't appear because this appeared a second time. I mean that may have occurred but I don't know but that's an issue.

TM: Right okay. Is it important enough to kill all those extra trees? And the answer with this one is probably yes. But there's ethical dimension in most of the things that we do.

DG: Right. So we're still getting comments back on version 2. We've got version 3. We're getting the ballots back on version 3. I have a student helping to collect the ballots. We're organizing stuff and getting responses. I contact the Steering Committee and I think at this moment I've forgotten who Frailey's colleague, IEEE colleague was on the Steering Committee. But I say, "This has all been done. I've been following the schedule. You have this December deadline." And I am told that "the committee may be going out of existence." The Steering Committee may go out of existence in December and put in a file all of those things that were not accomplished. So now you have this very simple pressure which seems to me if we don't finish the code by December all of this work is going to go into a file drawer and we'll hear, "Thank you anyway. We appreciate your effort."

TM: And you heard this as a rumor?

DG: This one came from Frailey that we had to move on quickly because. And as I heard this I went back to that comment in February of '95 at the meeting where he said to me, "The ACM may be pulling out" And I tied some of those together rather than it just being something about our committee not making progress because the other committees were not making progress.

TM: The one on education and body of knowledge.

DG: Education was waiting for body of knowledge to get done. And those people who thought linearly thought body of knowledge should get done, education should get done, and ethics should get done reflecting on this. And if we had let occur there would be nothing; if we had bought into that linear relationship. So now you hear from Frailey about this..

TM: Just a quick sort of fast forward. Those other two taskforces did they complete a significant body of work or did they...?

DG: Ultimately they did in the same way we did, which was to abandon the IEEE model and use our own model. That is what happens is, a group of, two very savvy, workaholic, closure freaks in Canada said they'll take on the project. That is they took it on and organized it and sent out a message and got people who would work and they worked through. And yes there is a body of knowledge that is a standard now. And that is a result of...

TM: And those two Canadian, control freak, workaholics do you know their names? Just so the record is clear because it's these three forces moving...

DG: I know. Pierre, Pierre is one of them who had nothing to do with any of the original stuff on the body of knowledge. The original stuff on the body of knowledge had interference with companies like IBM and other things, interference/support.

TM: We can fill that in later.

DG: Pierre Bourque and I've lost his name. [Robert Dupuis]

We're now in September/October when I hear this exciting news about December. So I tell Dennis if it's going to happen we have all these ballots coming in and we'd need to have a meeting. The "we" being the three members of the steering committee, Miller, Gotterbarn, and Rogerson, and we need some funding for that. And typical of the Steering Committee the very first debate was "Well how much is each society going to contribute?"

TM: Oh no!

DG: And then it became, "Well we'll split the bill 50/50." And you go, "Excuse me you're going to have billing coming from three people at different amounts and each society," I had to ultimately do the accounting like this. And it became one of those stupid things that said, "Well Gotterbarn spent this many dollars, this many dollars, Rogerson spent this many dollars, this many dollars. So if we add this portion of Miller's to this one and this portion of Miller's to Rogerson it comes out with a ten dollar difference at the bottom." But you needed to have receipts. So you had to figure out how to balance the receipts. One of the ways to waste five hours of your time is to make this work out, so it was within ten dollars.

We got the financial support. I arranged for a hotel room in Baltimore. I could reach Baltimore easily. Miller who was then in southern Illinois could reach Baltimore using Southwest Airlines cheap. Rogerson could get a direct flight to Baltimore from London. There's a hotel I had done some work in before. They had every room was actually a suite with a living room kind of area and a bedroom area. We met, got together and met each bringing laptops and I having a pile of document with all of the comments and every thing going on. And we sat in this room and we worked version 3 given all of the comments into version 4. The number of changes, we made changes in structure, changes in the grammar of some of the statements.

Everybody, now here's a thing that was discovered in the comments. What I had put in the preamble where I distinguished things into three different levels half of the comments were you've got to revise the code and distinguish which of each of the imperatives corresponds to which of the levels, and we knew that whatever we did in that kind of division would just generate new excuses for criticism and would not be productive. That is if we put something down as professional and not technical professional someone

would say that would belong in professional. If we put something in purely humanity level someone would say, “No it’s professional.” So we simply erased that and pulled that out of the code, because I would say fully half of the comments that responded negatively were about that.

It was heartening because most of the items... I’m going to have the wrong issue with the numbers but most of the items received better than 90% approval. And it was wonderful to see some of them come in. Some of them were upset because of the grammar of things and they would argue against one clause but the one clause about diversity got about a 70% approval. It was the only one that dropped below 90%. Now those people who said, “I strongly approve,”...

TM: Are you saying it was a question of people responding the kind of politically correct language that...?

DG: That was where we... We look at this and wondered who would object to a “Don’t discriminate” clause, because other kinds of thing, the preamble of the codes says you’re going to work hard for other human beings to try to forward human safety and so on, and then to say but you should discriminate doesn’t seem to be consistent. And so we looked at it and then we looked at the ballot and we kind of looked at each other and it’s the kind of moment where you hit yourself in the head for not having seen something that was obvious before. Miller and I looked at it and said, “The Americans seeing ‘political diversity’ would go, ‘What’s that doing in the Code of Ethics?’,” or something like that.

TM: Cultural diversity, wasn’t that your phrase?

DG: Cultural diversity, yes. And it was just didn’t... We realized, or we had a hunch that this is what the problem was.

TM: And how did you deal with that issue?

DG: We had this vote that it was still 70% but we rephrased the code in a couple of ways. I still learned a lot from the ACM process, which isn’t that we modeled the physical ACM but we avoided some of the errors we had made in building the ACM code. I was able to bring some of that experience to try and avoid some of those errors in this code. One of the errors in the ACM code was when we had written it we had a clause that said, “Don’t discriminate on the basis of religion, on the basis of sex, and on the basis of race,” and we had a period. Several of the comments that come in when the ACM code in the early 1990s had got out for ballot were that you left out discrimination on sexual preference; you left out discrimination on trans-dressing, you left out discrimination on... And what we realized is that whenever you put in a list, every new group as you begin to learn that you’ve been stupid. I mean in the 1960s we didn’t have, “Don’t discriminate on the basis of sex.” It was in the ‘70s with the women’s movement that they brought it to our conciseness that we’ve been pretty stupid before. So as you put a list in, at some point we will learn something better and you have to change the code. So what the ACM did was pull out the list and say, I’ve forgotten the way the clause goes but it’s something

like, “Don’t discriminate on irrelevant grounds.” We always make discriminations. I make discriminations about what kind of car to buy but I base it on cost and efficiency and so on, not whether the guy who put the tires on is a Nazi. Don’t make, don’t discriminate on irrelevant grounds or something like that. I’ve forgotten the line but something like that.

TM: Could I ask a reflective question? I see the procedural problem with enumerating the various classes of people that you’re not suppose to discriminate against and then sets the code up to be a kind of time capsule. You can see that it’s from the ‘70s or ‘80s or ‘90s or some other decade. But is it also the case that you say that by stating say a statement on sexual preference or some other criteria that you’re actually being more specific then the code would have greater bite? In other words somebody would say, “Irrelevant criteria, but I would never discriminate on irrelevant criteria,” although they might not be sensitized for the discrimination that unthinkingly they do.

DG: Yes, there’s this balance. Yes, that’s an issue. They way this one had been... I think we ended up shortening it in the way, leaving it open to that criticism.

TM: Not enumerating specific ones.

DG: Yes, not enumerating specific ones. You would have a tough time in today’s society saying that sex is a relevant issue, in which to do discrimination of hiring unless you’re talking about dancers at clubs or something like that. So those were the initial sorts of things. You started to look at the questions and it was a set of, a period of time. I do my standard model of “Let’s start and keep going until we get done” and that became a bit of a joke. But Rogerson became the scribe and Miller and I were going around working with the stuff and Rogerson working with the stuff and keeping track of all the changes. But then we started to look at the code...

TM: Rogerson was your scribe?

DG: Yes, Rogerson, Simon Rogerson was the guy who was sitting at the one PC. Miller and I joke about being old guys and we work better tearing up pieces of paper than modifying things real time on a computer.

And we discussed every question, and then there were those... Part of the process that went on was as you start to discuss every question one of the three of us as you get on to the third, or fourth, or fifth set of suggestions is noticing a trend of some sort and realizing that the code is leading towards a misinterpretation because of something that it wasn’t pointing to. So the original draft that Davis had done had started out with “Product” as the first thing because Davis had this thing in terms of what was most important in impact. And within about, I don’t know how the time went, but at some period in time we said, “You know, we’ve got one or two people who’d say, ‘You’ve got to rearrange these things.’ Someone would say, ‘Why is product first?’ And you’d say, ‘our point in this code is to say that as a software engineer the very first and most important consideration is the impact of what you’re doing on the public’. So we flicked

'Public' up into the first clause. And we started to restructure the clauses, and I've forgotten how the restructuring went, but we restructured it based on what we considered as we're reading the comments and as we talk among ourselves the order of these major principles, major categories.

Then as you start to look at the categories, and we're going over each one. And it was interesting to watch as an afterthought I mean there was thought in the air. It's a stupid comment but we'd read some clauses, we'd read some comments, we'd read some clauses, and one of us would click about something that had gone on before and say, "You know, this is the problem we had here and it was the same kind of thing there," and they'd see a relationship. And we'd restructure it, move an item down, change some of the wording and we did all of this restructuring in the light of these comments, in light of what we knew, in the intention of what was going on.

And we had another underlying motive. Working with the joint committee of the IEEE and ACM, Steering Committee, having them do something together was amazingly significant because the fact that those two groups worked together and produced a document of some sort meant wonderful stuff. We also had the view, given all the things that had gone on before, that this was not something that was likely to happen again since they would abandon ship in December. So we built a code of ethics and guide to professional conduct and we clearly articulated that this was both a code of ethics and a guide to conduct. So we had two levels of code in the same breath. Some of the people later objected to that but part of our motivation was this was the one chance we had and if we lost it then it was gone. And we went on for hours, just as you're probably experiencing this with me now, but when you're having fun and you're making progress...

TM: Yes, just as a point of fact it's 3 am and we've been at this for seventeen hours straight. It's not that bad. As a point of fact it's 5:48 and it's pm.

TM: You've also described, we can get this on take down, but a certain kind of real joy and amaze in a kind of secret accord between the three of you because it felt like you were doing something really significant and that your work really clicked together.

DG: I picked these people, no I'm taking credit. I was lucky enough to have met these people at interesting points. Remember Miller's name comes into the story in 1989. When in 1989 an article appeared in the Chronicle of Higher Education that named the three people they could find in the United States doing computer ethics; Spafford, Gotterbarn, and Deborah Johnson. Miller should also have been on that list; practitioner set of values. Rogerson when I first met him it was in relation to the early days of this committee, when Bob was still involved and before I ended up going to Rogerson's center.

In 1996 in July, I was at a conference in England, the Professional Awareness in Software Engineering Conference. I was presenting a paper on software engineering ethics and a report on what the Steering Committee was doing, so I had two presentations. Rogerson

had a paper and his paper was entitled “The Ethics of Software Project Management”, and he was bemoaning the absence of an ethical approach and the consequences of doing things and you could see... There was a synergy instantly between myself and the audience, and him up there. In fact I’ve been working on an ethics risk analysis process so when he was saying he didn’t know how to handle it we still joke about my arm shot up and I offered him an answer. And I said, “Do this process, do this analysis and so on.” We spoke shortly after that, after he presented his paper, and he invited me to his center, the next week brought me back over in an effort to bring me over during my sabbatical year. And that’s how the Rogerson connection was established.

But the interesting thing is that, is the set of things. Three people from very different backgrounds, the tie is that we’re very committed to a sense of that we’re willing to fit every moment of negative space to this project. I mean, the amount of time that these guys put in is just amazing. We’re committed to the same value set. Miller has taught other faculty members how to teach computer ethics. He’s still writing in the field. He still works with computers. Rogerson has the center. I have the center. Rogerson and I work on conferences in Europe on computer ethics. And you got together, and there was no question in the room; we had an opportunity, we had, we hoped the collective intelligence, we had the commitment and value set. It was really interesting because it was never an issue of, “You’re going the wrong thing,” in the discussion it was, “Maybe we can say it better this way. Maybe we can say it better this way.”

I’m the grammatical Philistine in the crowd. Miller has a wonderful sense of prose, Rogerson would bring in his English Literary background, and they would structure sentencings. “You’ve got to say this,” and I’d say it in some barbaric fashion. And Miller would say things like, “Let’s put it this way. If you have a judgment to make you consider first who it would impact, whether it can apply to the...,” and wonderful turnns of phrase in the preamble, which totally are his doing. Thoughts Rogerson and I... And it was a process where you did this for a couple of hours and you said, “Okay us as...” I mean, the fun part was us as old folks at five or six o’clock at night say, “We need paper copies of this to think about tonight.” We’d go to a Kinko’s to get it printed. We’d go with, at that time, floppy disks to the Kinko’s. They couldn’t get their printer to work and they said, “We couldn’t get it printed.” We couldn’t allow that. They didn’t know what they were hit with All of a sudden, “Well what’s wrong with your printer?”

“Well we don’t know we’re not very good.”

“Okay, we’re computer scientists.”

TM: We should be able to figure this out.

DG: And it was a stupid loose connection at the back of the printer but you know you did all the software stuff first because you thought they’d check the hardware. You print this out, for each of us, we had dinner and we went back. I don’t think we worked that night because it was, travel had been involved. But the next day we came back and in the morning and put another whole day in. And I think pretty much we had a wrap on it,

checked everything, looked at all this and we had version 4.0 at the end of that day. And we went our ways. It's really amazing because I look at Rogerson's description of that event during the interview and Miller's description of that event in a private letter to Michael Davis and it's, this was... You had just as you said three people, same set of values, all working to a common goal, and you knew that if we could get it together then all of this work might make a difference. And it seems to have.

So we went our way. Rogerson sent me the documents. I made a few other corrections. And it took a couple of days and I put together a message to the Steering Committee. "Here is 4.0. This has got the votes," told them what the votes had been, 90% for most of these things, "Here is the structure, here is what we have done, here is how we responded to things." I'm sorry before I sent it to the Steering Committee I sent it out to the ethics group again, and said, "Any comments. What have we missed? Have we got, is there any sense to it?" Some things came back and I made some modifications to it because yep, they were right and we missed something. Then I sent it off to the Steering Committee and said... Now notice it'd gone since that February now to December without them telling me what an approval process would be like. So I did the same thing again, I defined it and said, "We hope that you will approve this code and send it with your recommendation to the governing bodies of the ACM and IEEE Computer Society recommending their approval." Nothing.

TM: There's a pattern here

DG: I have a pattern too.

[end of tape 3 side B]

TM: Okay, we're on tape again. Don Gotterbarn being interviewed by Tom Misa, 22 September 2005. Don's summarizing this year.

DG: The year 1997 was an interesting year because it has so many lessons to be learned. One of them that I find as great fun is the number of errors and kinds of mistakes that got made. The original project started out in a fruitless direction to identify ethical standards in an ill-defined way, in a way that nobody really understood what was going on. Because of some confusion about the way in which technology operates, Ed Mechler took over the subgroup of the Professional Competence group, and put together what they thought was supposed to be a code of ethics rather than a list of standards. Michael Davis organized those into a useful structure. Both of these activities could be considered mistakes.

Further ahead into the politics of this was a significant political error of just violating the chain of command and somebody from a committee that nobody had heard of sends to the general of the Steering Committee final results; opening all of the political hatchets and knives and activity going on. That almost spelled disaster for the project. Again what was presented now as a code of ethics was now a deliverable. It was at this point the only deliverable from any of the taskforces that the Steering Committee had initially set up. Because people were committed to the code of ethics, other kinds of things went on, other

mistakes were made. Advocacy of publication in several journals and a potential disaster of having to put a statement at the front of these article that said, "It doesn't really mean anything, don't worry about it," was then responded to in a way again where it was fixed by a cover message. And I'm losing my theme because what I want to do is sum up the mistakes.

TM: You're doing the kind of narrative summary of all of the...

DG: Yes, I wanted to characterize them as mistakes.

TM: So you wanted to draw out some lessons learned.

DG: And so, one of the alleged mistakes was modifying the code in the ways we did in Baltimore because we made some significant modifications. The end result of that, set of taking license, which could be viewed as a political mistake, ended up to be quite valuable. Followed by the next set of mistakes. In December of that year the code was submitted to the steering committee. Nothing was heard from the Steering Committee. Gotterbarn did not know whether it was in existence or out of existence anymore. So given the plan that he had originally suggested, that of recommending the code to each of the governing boards, Gotterbarn choose... I sent the letter that I had sent to the Steering Committee to both the ACM Council and the Board of Governors of the IEEE saying that I had heard nothing from the Steering Committee. Quickly I heard from two people who had been on the Steering Committee indicating that I should not have done this, another political mistake, but what had been done was to make the code public again outside of the Steering Committee as an accomplishment going around any opposition on the Steering Committee.

The sets of mistakes that were made, my relying on a procedure to develop software standards and it having to fail, led to or made possible Ed's development of the code standards. Each of these pieces is interesting to me because I don't think the code would be where it got to at this point if you took away any of these so called mistakes. It's my belief that the ethics taskforce would have gone out of existence, still floundering wondering what ethical standards were with nothing accomplished. I'll go back to the narrative now. You look like you have a question.

TM: No, that's fine.

DG: We now enter a period of time where code is attempted to be approved.

TM: Now we move into 1998?

DG: 1998, I guess, yeah. And the code version 4.0 has been submitted and we're still receiving comments now on 3.0 that was published in the September and October editions of Computer and of the Communications of the ACM. Chasing down meetings of the ACM council gets to be quick tricky. Consistent with communications difficulties, two people who were on the Steering Committee, Leonard Trip, an ex-president of the

IEEE Computer Society, and standards director for the IEEE Computer Society and standards director at Boeing in Seattle, and Dennis Frailey of Raytheon and the ACM, have been working to continue the software engineering enterprise of the Steering Committee, which is now, unbeknownst to me, fully out of existence. Fortunately, there's a product on the table that is now quite public, which cannot be ignored. In several efforts to contact the committees nothing was successful. In May of that year, the ACM council was meeting in Washington DC. I arranged to have an item on the agenda related to the code and appeared in Washington D.C. at the council meeting. There was all sorts of confusion and misstatement, and all sorts of confusion going on about what was happening with the code because unbeknownst to me Frailey and Trip had submitted a proposal about the code which they had not communicated to me. I had submitted a proposal about the code that was for the ACM to approve the code.

My bullet item came up first. It was quite interesting because some people asked questions about, "Well, have you submitted this code to review?" and seemed to be unaware about anything that had gone on in the Communications of the ACM. Even more interesting was one of the members of council who had been on the Steering Committee also seemed unaware of the existence of the code. As the code was being presented people pointed to the upcoming agenda item...

TM: And you were presenting version 4?

DG: I was presenting version 4 for adoption.

TM: The Baltimore...

DG: The Baltimore...the slightly revised Baltimore version. There were some revisions after it came out of Baltimore. I don't remember what they were, but they were slight but respected. That was we said it was important to do it.

Then you hear from a representative from Dennis Frailey so the message is boggled, the message is corrupted, that there's some type of other review that needs to be done. The other review was to put the code through an IEEE Technical Standards Review. It was being suggested by Dennis Frailey and the ex-president of the IEEE Computer Society. I thought the battle was over and lost because a technical review for most items takes several years. I went back home quite deflated and called Leonard Trip who was the IEEE-Computer Society ex-president. Leonard explained the review process...

TM: You mean the IEEE Computer Society?

DG: The IEEE Computer Society. Leonard explained that he had a review process in mind. I'd never worked with Leonard before. It was a little bit scary because I didn't know whether, I had no understanding of his politics and his battle. I had pretty much scoped out everyone else's politics and that's why, in part, the code got to this stage. Now there was this voice from Seattle that was quite organized, quite businesslike and said,

“No it will not take five years. This is what I have a plan for,” and it would take a matter of about five months, or six months to do.

TM: This is the IEEE Standards Review?

DG: The IEEE Technical Standards Review. And the review process was a fairly interesting process where they ask for voters, people who want to vote on this. Then they have a way of making a balanced voting pool to make sure that it isn't all of the employees of a particular company, or all of the employees for another company, or all of the people from one country, and so on.

TM: Because of the problem with technical standards having direct proprietary commercial consequences.

DG: Well, it could if you built certain things into it. There's some legal stuff that we'll go into a separate discussion on.

So this is... So he asked me for people who I would recommend for this. So I sent him email lists of people that I had in an ethics list, sent him a list of the then ethics taskforce members which still included some of those people who had not contributed at any point. Nevertheless I was resistant to take them off the list. The list of voters got generated. The code procedure is that the code goes out and you vote “Yes”, “No”, or “Yes with reservations”. If you vote “Yes”, you've voted. If you vote “Yes with reservations” it means you voted “Yes” but you're making some suggestions that might or might not be followed. If you vote “No” you had to state exactly the reason why you voted “No” and what would lead you to vote “Yes”. We got back approximately 136 comments.

TM: From this Technical Standards Review?

DG: From this first level, layer of the Technical Standards Review. The next step in the process is to go over all of the responses, modify the code as the committee sees fit, and send it out to the voting group again. Those people who voted “Yes” could vote “No” if you changed the code in a way that offends them. Those people who voted “No” could say, there were some people who if you did exactly what they said in their “No” vote, their vote was automatically changed to yes. And those people who you did some of it or explained why you didn't do it could change their vote to “Yes” or leave it as “No”, but it's an up or down vote at this stage. So you now have comments some of which are impossible. They come from a committee where they'll say, “It has too much normative ethics. It has too much applied ethics. Now I'll give you contradictory comments.” So you can't do anything with them. I received these and this is now September, I think. And I went and started the school year and in this negative time I went through every comment and wrote a potential response to each of them.

These comments are by the way generally all put on the web. I violated the principle of this voting process because I called the IEEE and I said, "Please don't put these two people's comments on the web," because they had not realized that their comments would go on the web. And one of them in discussing why he had voted a certain way said something like, "My wife works as treasurer for this church and her church is using a pirated piece of budget software," and I thought inappropriate to reveal and the person didn't realize the consequences of it. And there was a similar type of comment. We put the votes up but restricted some of those non-relevant asides. Otherwise it followed 100% the IEEE standard. Now, I rolled out...

TM: I'm just double checking, you said you had approximately 136 comments and then each person would have an option for making a second round of comments?

DG: What happens is...

TM: There's structured procedure...

DG: ...we respond...

TM: Some "no"s are automatically changed to "yes"s. So it's a structured...

DG: And the final procedure is you vote. I don't remember if there's a comment place but you vote. And there's not a questions it becomes up or down. And this is very nerve racking because some of the votes you know you can't possibly change because they're by contradictory desires. Other votes, the two from Microsoft you know you're going to refuse to go along with because they say testing is the responsibility of the customer and you ought to delete the testing clause. And so there are those kinds of things and you have a limited approval votes and you need to be at a certain percentage. So you carefully evaluate all of these and you try and give, and you have several categories of response that is "I agree with and this is what I'm doing", "I partially agree with and this is what I'm doing", "No I don't agree with you at all", and "Tough". So I wrote out this set of responses and fired them off to Miller and Rogerson, and I took a set of comments.

Now another one of those meetings occurred. In the stuff Rogerson and I and others had been doing, we were helping to organize a series of ethics conferences called Computer Ethics for the Philosophical Enquiry and that was having a meeting at Dartmouth University. And Miller, Rogerson, and Gotterbarn were presenting papers there. Terry Bynum, who I mentioned earlier around this 1991 thing, has been active in computer ethics all the way through, and he was on sabbatical leave at Dartmouth University and had this room above a barn, two or three miles away from the university. Well, you got the trio together and we took all these comments and we went out to this room above a barn late one evening and Bynum... And we went through each of the response. I have a style of asserting things emphatically, enthusiastically and I rely on Miller and Rogerson to say, "Calm down Don. Say it this other way." And one of the reasons we work well together is we all respect each other's contributions and it fits that way. And we went

through every one of these comments that I had made, modified them so that the three of us had agreement in modifying the code. The scary moment came...

Then there was a conference the next day in Boston just a little south of Dartmouth that was a philosophy conference where I was giving a talk. When I got home within a matter of days I had the document that we had agreed on as a response. I sent it in to Leonard Trip and his organization that does this stuff. And they put it out for the ballot again. At that point you sit and you wait having no idea whether you have totally offended the world and they're all going to vote negative, or what's going to happen.

If "Yes" voters don't respond, it's considered that they're still voting "Yes". "No" voters can have an automatic turn to "Yes" because you've met their requirements.

TM: If their conditions are met.

DG: If their conditions are met. And you can assert their conditions are met but the people at the IEEE looks and say what they wanted, and what you did, and they determine whether you've really met their conditions. Then other "No" voters if you said, "We can do this but we don't want to do it this way because it may allow this interpretation and that's why we wrote this thing this way." Or, "What you want here is really handled in the preamble in the second clause and this thing is long enough already and we don't want to add a second thing." Those were the kinds of arguments you used. Sometimes it was flat out, "No this is unacceptable to a software engineer."

And I've forgotten how long after the vote came in but the vote came in with, "Yes it passed." And it passed by, I'm not going to say anything more. I remember that it passed and it passed without it being a close call.

TM: The figures are part of the record somewhere else?

DG: Yes. It's actually all of this is available on the web still at this moment. You can read every one of my comments. You can read every one of the issues. You can read the final vote. This is IEEE procedure, every thing is wide open. And that's a useful procedure I believe.

TM: So all of your comments...

DG: All of my comments as modified by Miller and Rogerson. So I have to be careful to not take credit that I'm the author of whatever great prose might be there. It's the committee's comments are there. I authored the first straw man of each of those and they modified them as necessary.

TM: If you went into the IEEE's website is there an obvious way of navigating to this or is it for members only of IEEE?

DG: You can get to it but I don't remember how. You'll send me an email and I'll try and get it to you. Or I think it may be in Davis's book some place.

TM: Because this is a nice level of documentation that somebody's whose following up on this history might want to know the comments from somebody, might actually get the raw comments. That it's also a kind of secondary archive that would be nice to know.

DG: So now you have this vote and you all of a sudden. Not all of a sudden, in further comments with Leonard Trip you realize that he's the standards man and he would be happy if this passed. And you realize that he wasn't against you. You pick up the politics a little bit later. That is was in fact an incredibly savvy political move that I had awareness of how smart it was. That's probably the biggest political stupidity I had in all of this. But while I'm scared, I did everything to the letter on this thing.

TM: It was savvy in what way?

DG: Well, now you've presented to the ACM again and you are asked, "Well, what makes you think software engineers accept this code?" Your response is, "It went through the IEEE Technical Standards concept and passed the most rigorous form of testing currently available for a technical standard. " So it's not wimpy. It's done and there is no question about the evidence.

TM: Have you any idea what the Technical Standards people thought about this, because this is not a standard technical standard? I mean it has nothing to do with the inside architecture of a chip or a wireless standard. This is something else.

DG: I have no idea. The one thing that it was, is an accepted procedure for validating a document, and proving that the community of developers accepts it. So what this was not a vote that Gotterbarn was waving his hands about. This is, if you want to count the votes and see the comments you can go to the website today and read every one of them. I mean the brilliance of this, of forcing me through this, and following to the detail, meant when the ACM said, "Does it meets software engineering standards?" you go, "It meets it."

Then they ask, "Well how do we know this whole thing is consistent with the ACM Code of Ethics?" They say, "We need to have the Committee on Professional Ethics review it." And then I agreed to that because I was the chair of the Committee on Professional Ethics, Miller was one of the members on the ACM Committee on Professional Ethics, and other people who were involved in the process where also on the Committee on Professional Ethics and we know that it would pass. The only problem was someone on the ACM said we need a grammarian to review it because he was upset with a particular split infinitive. And we said okay and took care of the one split infinitive that was identified.

It then went before the IEEE Computer Society and began... Leonard Trip had put them the same bind. "You are the IEEE Computer Society. Here is a code of ethics that has gone through your strongest procedure. What can you say about it?" And what happened

is it was approved by the IEEE Computer Society with one caveat. They wanted their lawyer to review it to see if there was any legal problem with them supporting this code of ethics. And you're going to go in my failed memory because yes, there was one sentence that required either a semicolon or something very trivial so that it wouldn't be subject to a very perverse interpretation that the lawyer had. I'm sorry; my attitude is coming out there. It wouldn't be subject to one of those alternate, possible interpretations you could give of what seemed like a clear sentence to everybody who voted on it. And it was interesting because it was one of those that received no objection. That being done I sent a letter to the IEEE Computer Society saying... No, I called Leonard and told him, "Here's what we've done," and sent him the revised copy of the code, and he carried it through to the Computer Society and the code passed.

And now we have a code for the IEEE Computer Society and the ACM. At this moment I develop a concern. I know the ACM charges people for copies of publications, charges people to do certain things, and I have this moment of panic and fear going, "Uh oh." So I file a copyright registration on that code and pay my \$20 out of my very own pocket for the copyright in the names of Rogerson, Miller, and Gotterbarn as copyright owners of the code. Then we have the next minor battle which is, since this code has been adapted, it's time to print the final code. And time to print this final multiple-voted on version, multiply reviewed version in the Communications of the ACM and IEEE Computer Society. Calling each of them as I had done before saying, "There is no committee to say go ahead and get it published on..." Saying, "In accordance with my plan, back in February, not rejected by the committee, it's time to publish." And I'm told, "Okay, well you have to fill out the copyright release forms."

And I say, "No I don't. It's copyright and I want people to use it."

And you hear, "Well, this was a work for hire."

And I say, "Excuse me, a work for hire requires pay."

TM: You got no pay. Ha ha ha ha

DG: And you go... And there's some mumble, mumble, mumble. And then you call the IEEE Computer magazine and you get a similar kind of thing. And I say, "Here's the deal," and I communicate with both of them, "I will sign the copyright," I mean this is one of those really interesting things because I'm not sure if it's legal. "I will sign the copyright over to both magazines, to be held jointly by the ACM and the IEEE Computer Society, provided that you put at the bottom of this document that this document may be reproduced entirely, can be reproduced in its entirety without cost or the need to request permission if this copyright, joint copyright ACM-IEEE, is included."

TM: That was your fear of the ACM charging, because otherwise they would charge...

DG: Otherwise they would charge... People who wanted to publish it in a book would have to pay to put it in a book. One of the easiest ways to get it distributed is to say, "Hey, it's for free!"

TM: That right,

DG: There was another thing and I guess we have to go back to the Baltimore meeting.

TM: The Baltimore hotel meeting?

DG: The Baltimore hotel meeting and I'm sorry to do this. One of the major decisions, and probably the most disgusting one is it at the Baltimore hotel meeting, I'm sorry my memory just popped up like this, was that several of the comments said the code was too long. To keep that whole thing in mind, it was too much. Well the code was structured with eight basic principles and clauses under each principle. So we said, "Well we could develop a short version of the code with just the principles." So we re-worded some of the principles so that they could almost stand alone. And we built the code, changed the code so we put a short version in the front with a short preamble to the short version, the longer preamble, and the rest of it behind it. But the three of us were very anxious, and I guess the only real concern we had was as soon as you made a short version, someone would only pay attention to the short version. So the copyright notice says you can publish it with permission and without cost provided the code is published in its entirety. Also in the preamble of the short version, in the preamble of the short version it clearly points you to the longer discussion and says this preamble is just a reminder and pointer to the real issues.

TM: OK

DG: Both the ACM and the IEEE Computer Society accepted this thing. And that was another one of these things, I'm not sure of the legality of this and they questioned the legality of it but I know it's been published in multiple textbooks and that it's appeared in articles. And I think in part if you're a publisher and someone says, "We published this code," and the question is, "How much does it cost?" And the answer is zero. You don't have to pay some book and everyone where it's published. It's easy to publish it. It's easy to put up on your website. So that got accepted and the code got published. And that's how the Software Engineering Code of Ethics and Professional Practice came into existence with a million little side stories that I've left out.

TM: Was there a certain moment in time where you thought, "Okay, this is it. This is time for a celebration."?

DG: Oh, several of them, some of them mistaken. When we came back from Baltimore, euphoric is the best way to describe my attitude. It was, "We've done it right. We've met the objections. This is done exactly what I've hoped to happen from the beginning. And all the years of silliness from '93 to '96 until the world got rolling. It was okay." And you've said you've met the deadline and you've done this. And then of course this

elevation collapses as December and January goes by and you're trying to contact these people and nobody is responding. And then it goes worse into the pit when you go through... That next year was a real downer because of all these comments coming in and you were never really... The IEEE vote worried me because any fickle thing, having made the mistake about 'cultural diversity' in that first one and not realizing it you go... Because all it took was you to object to one clause in the code and you're vote was not "I object to this clause", it's "yes" or "no". If you objected to one clause and said "no" because of some stupid structure to it that we didn't realize we were dead.

TM: If ten people... If Person 1 objects to a Clause One, Person 2 objects to Clause 2 but only Clause two, Person 3 objects to Clause 3 but only Clause 3. Those all get registered as "no" votes?

DG: That's correct.

TM: So there's a kind of big barrier against getting the body of it passed.

DG: The... Yes. And one of the things that deceptive is, I've forgotten what the number is whether we got 60, 70 or 80%, but when you see the number at first and you sit down and look at it and go, "That isn't as impressive as the 90-some odd percent." And then you look at the "no" vote is because of a particular wording in a particular clause and they loved the rest of it. And it's a very steep hurdle to get over and when you get over it... It has the problem of when you look at the number and you go, "Oh, that isn't the 90%, that's 70%." No, that's 70% said every clause is perfect, 30% had some objection somewhere.

So I guess after the December's launched the code out, I sent out a celebration message and then the vote goes, "Uhhh." And then you communicate with the guys again, with the people on the ethics taskforce again, at some point where it's past the vote. "Hurray, we've made it," but then you realize you haven't gotten the committees, you haven't gotten the ACM Council and the Board of Governors of the Computer Society saying, "Yes."

TM: The Technical Standards Review, the way you described it, was a really powerful lever on both of those groups.

DG: Right.

TM: Especially I would say, especially pressing on the IEEE CS because this is their Technical Standards Review, but also that must have made it certain that the ACM couldn't dismiss this as being a whatever.

DG: But you have to presume that rationality will prevail in the decision. And I won't say anything about either of the committees, but sometimes things can go crazy for no reason at all. Some... We have had several companies that have adopted the code. They have them as conditions of employment where you read the code and sign it and say you'll

follow it. We have one company in Florida that was going to adopt the code until their lawyer looked at Clause 6 point something which says follow the laws of you're whatever unless a significance issue of conscience arises that will hurt society or something like this. And that lawyer said, "We can't accept the code because it says there can be occasion where you can violate the law and we're a company and we can't let you violate the law." So they have their own version of the code that has everything except that particular clause. That's all it takes.

TM: That would have been a "no" vote in the IEEE review, if it were the same structure.

DG: But I'm talking about on the council. I mean one of the things that got very awkward with the ACM Code at one point, this early code which finally got approved in '92. There was a meeting we had where we were discussing it in Dallas and someone from the ACM was at the meeting who was a lawyer or thought themselves a lawyer. And all of a sudden all sorts of things about liability the ACM would have if they approved this code, because it would be certifying that all ACM members followed this code. And you had to then deal with that because if that occurred at a council meeting and he convinced some people at council that that sort of position was a true legal issue, they would have worried about it and closed things down. So you never know what's going to happen at any of these meetings. And I was not at the IEEE Computer Society one so you did not know what was going to happen. That got the code to be an official document.

Now lessoned learned; keep people working, don't step on political toes, and if it's necessary there are some names of people who were on the taskforce on the Software Engineering Code of Ethics who probably did nothing except having their name there made it possible for it to get passed. So they contributed in a funny way whether they'd like to it or not. There's if you can keep a good handle on all of the political issues, now I may have some allusions about what political issues were going on but by taking strategies to cover all of them if they were or weren't going on they were addressed. And then given all this planning I'm sitting here talking about, the code wouldn't be here if it weren't for a set of mistakes. I've articulated what they are earlier. And it really is a strange process to look back on because you go, "Well I'm not a good politician. I'm in people's faces all the time. And yet I managed to get this sucker through." And how that happened is because the only smart thing I did was to select some good people; Rogerson and Miller and some of the others. And others emerged that I didn't select. And I'm out of speed at the moment. You want to stop and we'll think about what I should be talking about next? Or what you want to ask?

TM: yes.

[End of Interview, tape 4 side A].

© Don Gotterbarn