Oral History of Watts Humphrey

Interviewed by:
Grady Booch

Recorded: June 17 - 22, 2009
Via Skype

CHM Reference number: X5584.2010

Computer History Museum Oral History

for

Watts S. Humphrey

Topics

Boeing; Team Software Process (TSP); Personal Software Process (PSP); Software Engineering Institute (SEI); Microsoft; Intuit; Embry-Riddle University; planning; Department of Defense; National Medal of Technology; agile programming; SCRUM; MOBIDIC; IBM; OS/360; FORTRAN; FAA; Watson, Thomas J, 1914-1993; Poughkeepsie; Endicott; RCA; SABRE; IBM System 360

Contributors

Booch, Grady, Interviewer
Humphrey, Watts S, Interviewee
Spicer, Dag, Editor

Publisher: Computer History Museum

Date: 2009-06

Place of Publication: Skype

Extent: 177 p.

Collection Title: Oral Histories Online

Accession Number: 102702107

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

The Oral History Background

The Watts Humphrey Oral History

Dag Spicer, Director of the Computer History Museum in San Jose, California, arranged for Grady Booch to interview Watts Humphrey. This document is an edited transcript of that interview. Note, however, that the text exactly follows the interview except of minor editorial corrections to correct obvious transcription errors.

In this oral history, Mr. Humphrey describes his early life and formative experiences as well as his employment history, first as a senior manager at IBM, then as an early member of the Software Engineering Institute. Humphrey covers a wide range of topics and companies and the software challenges they have faced. Humphrey also describes the Team Software Process and the Personal Software Process (TSP and PSP) methods for developing quality software.

Watts Humphrey Biographical Information

Watts Humphrey is a Fellow at the Software Engineering Institute which he joined after a 27-year career at IBM. He authored the original version of the SEI's CMMI process maturity framework and developed the Personal Software Process (PSP)[1] and the Team Software Process (TSP). He holds graduate degrees in physics and business administration and is a fellow of the Association for Computing Machinery (ACM) and the IEEE. His publications include many technical papers and 11 books. He holds five US patents. In 2005 at a White House ceremony, the President of the United States awarded him the National Medal of Technology. This is the nation's highest honor for technical achievement.

Grady Booch Biographical Information

Grady Booch is Chief Scientist for Software Engineering at IBM's Thomas J. Watson Research Center and was Chief Scientist of Rational Software Corporation before its 2003 acquisition by IBM. He is recognized internationally for his innovative work on software architecture, collaborative development environments, and software engineering and is one of the original authors of the Unified Modeling Language (UML). He is author of several hundred articles on software engineering and six books and he originated the terms and practices of Object-Oriented Design (OOD) and collaborative development environments (CDE). Grady is an IBM Fellow, an ACM Fellow, a World Technology Network Fellow, a Software Development Forum Visionary, a recipient of Dr. Dobb's Excellence in Programming award as well as three Jolt Awards. He was a founding board member of the Agile Alliance, the Hillside Group, and the Worldwide Institute of Software Architects. He also serves on the advisory board of the International Association of Software Architecture, is a member of the IEEE Software editorial board, and has conducted several oral history interviews for luminaries such as John Backus and Fred Brooks. Grady received his bachelor of science from the United States Air Force Academy and his

[1] Personal Software Process, PSP, Team Software Process, and TSP are Service Marks of Carnegie Mellon University.

master of science in electrical engineering from the University of California in Santa Barbara, CA.

Day 1 AM, June 17, 2009

Beginnings

Grady Booch: So welcome. This is Grady Booch. I'm here with Watts Humphrey and we're connecting via Skype because we're both in different parts of the world. Here we are on a morning of Wednesday, June 17 for an oral history on behalf of the Computer History Museum. So welcome Watts. Thank you very much for joining me. It's been a long time since you and I have seen one another face to face, so I'm just utterly delighted to have the opportunity to converse with you, so thank you for joining us.

Watts Humphrey: I'm delighted and honored to be interviewed, and particularly by you. It's been 20 years since we were last together and I'm delighted to meet with you again.

Booch: Twenty years. Oh, my goodness. Time moves quickly, doesn't it? Wow. Of course I have as much as hair and I'm as virile as I looked 20 years ago,

Humphrey: I can see that.

Booch: So let's begin. I'm going to start off with just some general philosophical questions here and then want to dive into just your life and how you got to where you are and then we'll end up with where you think the world is going. You're often considered by many as -- I think you're called either the grandfather or the father or the godfather of [software] quality. There are various monikers given to you and, indeed, you've been honored by the president by receiving the National Medal of Technology back in-- that was 2003, was it not?

Humphrey: It was the 2003 medal. They actually didn't award it until 2005. There'd been a few things going on then and President Bush didn't have time but it was a marvelous event.

Booch: Good and we'll talk about that. I mean, you truly have made a seminal difference in the art and the practice of software engineering, especially with regards to elements of quality. So I'm just going to ask a very broad question at first because this has been your life. What is quality and how does one measure it and how would you rate contemporary software systems in terms of quality?

Humphrey: Well, a very general response. Quality is doing the job the way it's wanted. I agree with Deming's and Juran's and the other quality guru's definitions. Quality is doing what it takes to truly satisfy the customer.

<audio break>

Booch: So here we are back again. Thanks to technology we got interrupted a bit but as I was saying earlier, Watts, life was a lot simpler before we had computers, wasn't it? But my question on the table

was, you're really known as the father of software quality and my general question to you is, what is quality? How does one measure it and how would you characterize the quality of contemporary software?

Humphrey: Okay. Well, as I mentioned before quality is fundamentally doing-- producing what it is that the user both wants and needs. They're not always the same and that means in terms of cost and schedule, the way the product you deliver works. It's defect free. It continues to function properly and it's a satisfying product and that means we need to really understand what the ultimate customer wants. This is very consistent with the views of Deming and Juran and the whole quality community. I think that's a very standard definition and my reaction to the way that the software community has responded to this is that it's dismal. I mean that, by any measure for any other kind of product, software quality is truly terrible.

Producing quality software is a very difficult and a severe challenge. When you consider the quality of software itself and what it takes to make a high quality product, defects per 1,000 lines of code is one rather simple measure. It's simplistic but quite useful and five defects in 1,000 lines of code is considered poor quality, at least by me and most quality people. You want to talk about defects per million. However, when you look at software quality in human terms, five defects per 1,000 lines of code is extremely high quality. A thousand lines of code, at least my C++ code, was 30 to 40 pages of listings and five defects in 30 to 40 pages of listings, when you consider it, that's tough. And if we want to get not just 5 per 1,000 but 5 or 10 per million, we've got to talk about 5 defects in 30,000 to 40,000 pages of listings. The problem is that the whole software community is treating this as if today's products are high quality work. I mean, they're really, kind of, looking it over and saying, "Yeah, it looks okay." The real focus on producing extremely high quality isn't there. We're not going to get very few defects per million lines of code by just having people try hard and hope they're doing it well. The lack of a really serious attack on this problem by the entire community, the academics, the theoretical types, industry is I think, totally irresponsible. People need to understand that poor quality software work is responsible for just about all of the problems in the software business.

Booch: You used a phrase I'd like to dwell upon which was the human factor. What's the human cost of bad software?

Humphrey: Well, it's a terrible job. People are unsatisfied with it. They're delivering products late at excessive cost. The customers don't like that. Their management doesn't like that. Software people are really great people who are doing superb work, and they are treated like failures. They're failures on the job. I mean, this has been the history in software. You can't run it. You can't do a job in software and come out as a hero. Software people don't do that unless they end up building their own company and so we're failures. These are some of the brightest people on the planet and they end up failing. And I think it's-- the failure is in the way we teach them, the research we've done, the background and support we give them and the methods they use. I don't think we're doing a proper service for the people that we have and the big frustration I've got-- I can talk about this later. We'll probably get plenty of opportunities.—It is that we've come up with some ways to attack this problem and they're not getting any real attention by at least the research and academic communities. People don't recognize it as important and that's very frustrating.

Booch: In fact, I want to be certain as we get toward the end of the conversation, do remind me, I'd like to understand your perspective on what we, as an industry, could be doing to make a difference. But I'd like to turn now to how you got there because you have a lifetime of experience that lets you speak with authority on these matters but you didn't just get born this way and I'd like to explore the journey that got

you to where you were and the things that excited you about the field and drew you into it. For example, I knew you were a wrestler back in college and just curious, how does an ex-wrestler come to this particular domain? So let's start at the very beginning because these are parts of your history that I wasn't able to uncover. Where were you born and when were you born if you don't mind disclosing your age. I think you're 29 or something if I'm not mistaken but--

Humphrey: Thirty-nine. I've been 39 for almost 50 years now, right. I was born on the Fourth of July, honest to goodness, in 1927. That's the year Lindbergh landed. So I'm 82 in another couple of weeks. So I'm an old guy and my dad was an engineer, an MIT graduate as is my son, by the way. My dad's name was Watts Humphrey as was his father and he was a marvelous engineer. He was, as I say, a mining engineer and actually worked in South America. Jobs weren't available when he finished college in the early '20s. He was World War I pilot, by the way and in World War II. He was in the eighth Air Force in World War II. When I was born, we moved to New York and he got into the finance business. He was in the stock market for years and was actually a treasurer for a big insurance company. I think it's now what turned into AIG ultimately but also he had a marvelous job all through the depression. We were far from a needy family. We had a big house in Connecticut. He had a private airplane and all kinds of stuff, so I was, kind of, born into the lap of luxury. In first grade I essentially failed. I was a failure as a student. It turned out I was dyslexic and no one knew what dyslexia was and I didn't even know what it was until the 1960s.

I read an article about why I couldn't read as a child. I literally couldn't learn to read and I struggled and struggled and the school essentially failed me. They said I was doing terribly. So dad pulled me and my brothers-- I have two brothers-- an older and younger brother-- he pulled us out of school and moved us up to a little town called Litchfield, Connecticut. He insisted that I didn't fail, the school failed and he was going to get a school that would help and he did. It was a wonderful little school. It's become a school for learning disabled kids but it wasn't at the time. I had one-on-one teaching and I learned to read by the time I was nine and that convinced me of something that was, kind of, amazing and that is, I was-- as I say, failing but I found if I worked hard I could overcome it.

So I ended up being valedictorian of my class and all that kind of stuff and I was head of everything. I didn't do well with languages at all but that was always a struggle but other than that-- I mean, I was even editor of the school paper and so I just decided 'dive in and do it.' My dad's view was, "If you want to try it and you're willing to work at it you can do it." And that really changed my life, so I really did turn into a success going through that and it was a marvelous education. I mean, I really learned. I stayed at the same school all the way through high school and then went on. I was admitted to MIT with a scholarship and at Cal Tech with a scholarship and then I went in the navy.

Booch: Wow, I didn't know that, either. So what city were you born in? I didn't catch that at the beginning because I knew where you moved to but I don't remember the city.

Humphrey: I was born in Battle Creek, Michigan.

Booch: Oh, that's where Kellogg was from if I'm not mistaken.

Humphrey: I don't think we were there a year. Then we moved to Port Washington, New York briefly and then to Bronxville, New York. My next door neighbor was Billy Rickenbacker. His dad was the old Eddie

Rickenbacker who-- of World War I fame, started Eastern Airlines. Yeah, Billy Rickenbacker taught me to ride a bike.

Booch: Now, there's a great story. That's wonderful.

Humphrey: The Bronxville School was where I failed first grade and we got pulled out of there.

Booch: And you had two brothers, you said.

Humphrey: Yeah, an older brother, Phil, who is now in-- he's been a curator of birds for the Smithsonian. He's an ornithologist, zoologist. He's retired. Unfortunately, he's very ill. He was hit with a bad stroke several years ago and so-- why he was a first class jazz pianist. Unfortunately, he can no longer play the piano which is really a tragedy but he's a great success. He actually discovered the-- I think the first new species of duck discovered in 100 years. He was on the front page of-- second section of the New York Times a number of years ago and he's got-- that's there in the museum-- he discovered the Steamer Duck in-- down in Argentina. As a matter of fact, this last Christmas I took six of my grandsons on a cruise to the Antarctic which was a marvelous Christmas and some of the people there, they came up to me and one of them in particular said, "Are you related to the famous Phil Humphrey?" And-- oh, he was all excited because my brother used to go down to Argentina all the time, Tierra del Fuego, and that sort of thing.

Booch: My best friend in high school, his name was Phil Humphrey. Very different Humphrey, different line but he's now, I think, a lawyer in Texas. But I'd imagine if we dig way back there's probably some connection.

Humphrey: Oh, could be. Could be and my younger brother--

Booch: As you were-- go ahead. I'm sorry. Go ahead.

Humphrey: Yeah, my younger brother William actually went to MIT briefly and then he was in the Army of Occupation in Japan at the end of World War II and so he was working in Boston in electronics. He's an electronics design guy and he married a New Zealand lady. That's a long story that I won't go into but in any event they ended up moving to New Zealand and so he's in New Zealand now and he's been there oh, for 20, 25 years more-- no, about 30 years.

Booch: So as a teen growing up what were your passions and you learned to ride a bike from Rickenbacker which is pretty amazing. What were your passions as a kid?

Humphrey: I liked to build things and I remember my dad bought me a little Meccano set, a very beginning Meccano set. I couldn't have been more than about six or seven and he made some comment like, "You build all of that, and I'll get you a bigger set if you want it." I said, "Oh, great." So I did. My dad was working in New York and we were up in Litchfield, Connecticut at the time and he was commuting back and forth by airplane. He'd fly a little seaplane-- an Aeronca Seaplane-- from the East River and he'd fly it up and land in a little lake near us and he'd circle the house when he got up there. And we had a chauffeur and a cook and the chauffer would race out to the lake and canoe out to pick him up on a raft where he tied the plane. Then he'd come home and have dinner and we'd-- so he would commute back

and forth in the summer. In the winter he couldn't do that but I learned to fly before I could drive as did my brothers. So yeah, that was quite a life.

I guess I lost the train of where I was on the Meccano sets. I got so excited about building this stuff that I'd build everything and demonstrate it to my mother who would check it off in the instructions. I got through all the examples in the instructions in a week so dad said, "Okay," and he got me the next one. I said, "Well, what'll you do when I build all of them?" And he said, "I'll get you the biggest." So I went through all of them in quick order, built everything in every Meccano set and I got all the way to the last one and he even got me that one. And I said, "What'll you get me if I build all of that?" And he said, "I'll get you a car." So I couldn't believe that. I must have been seven or eight at the time and it took me a while to build all the examples, but I actually did. I finished it. He bought a Model T Ford, an old run down Model T for ten dollars. Can you believe it? What would it be worth today?

Booch: Wow. That's incredible.

Humphrey: We had a farm. It was an old farm back in an area near a trout fishing club that my dad belonged to. There was a farmhouse there and we'd all worked on it to fix it up and we'd all participated in ripping things out and trying to kill the many wasp nests in the walls. We put the car out there, so I never got to do anything with it but I did learn to drive it. Driving a Model T was exciting. I loved to build things and after that I got very interested in model airplanes and decided to be a aeronautical engineer.

When I was in high school, my dad went into the army Air Force.

Booch: What year would have this have been, Watts?

Humphrey: This would have been 1941, right after Pearl Harbor. I should mention that he had take a summer off in '37 and took us all to Europe for the summer. We shipped our car over on the boat and we drove all over. Through the low country, Netherlands, Belgium, down through Germany. Then into Austria, Switzerland, Italy and back up through France. We stayed in England for a couple of weeks and then came back home. We all had cameras and took pictures of everything. Unfortunately, when we drove through from the Netherlands we landed in Rotterdam and took pictures of everything and as we went through we took pictures of all the fortifications. It was kind of exciting looking stuff and when we got to Germany we pulled out our films to get them developed, but we never got them back.

Booch: How sad. They were probably considered some important war time kind of thing. I mean, you saw Europe right before the war then.

Humphrey: Oh, yeah, so we contributed to German intelligence I'm afraid.

Booch: Before we get further on this I do have to ask one question. Do you still have any of those Meccano sets from your childhood?

Humphrey: No. We moved so many times. My dad went into the Air Force in 1941. In the previous summer he drove us out through the west. He'd been told in '39 or '40 that he had six months to live and turned out he outlived the doctor by about 30 years. It was his medical problems from when he worked in a gold mine in South America right after MIT. In 1944, he'd been in England for a number of years with

the eighth Air Force. He came home in the spring and told my mother that they were going to have to get divorced. He had met a lovely lady in England and he basically divorced my mother and married her. When I graduated from high school in '44, I was 16 and turned 17 that summer. We drove out west and my mother went to Reno for the divorce and I and my brothers worked on a dude ranch in Wyoming. Then we flew out to L.A. My mother decided to move to Pasadena.

Booch: I have ask-- a small world story in Wyoming because my wife's family owns a dude ranch up near that area, an area called Glendavy. Got to ask, where were you in Wyoming?

Humphrey: We were at the A Bar A Ranch which was right by Laramie if you know where that is.

Booch: Oh, yeah, I know exactly where that is. Oh, my gosh. Small world. Yes.

Humphrey: Yeah, the Fourth of July I remember we were having trouble getting over one of the mountains because it was snowed in. My older brother met his future wife there at the ranch. That's where she was working for the summer. When we got out to California I planned to go to Cal Tech. To get in, I had to take two tests, math and chemistry. The math I had no trouble with, but I had never had a chemistry course, so a chemistry test to get into Caltech would have been tough. My older brother had been at Amherst for a year and he loaned me his freshman chemistry text. I spent all my spare time in the summer studying chemistry. I didn't have any lab work, of course, but by the end of the summer when I got out to Cal Tech and I took the chemistry and math exams I got a scholarship based on the results, so I must have done all right.

Booch: Incredible. That's wonderful.

The Navy

Humphrey: So I was ready to start Caltech and one day I stopped by a navy recruiting office and they talked me into enlisting. So I enlisted in the navy instead of going to Cal Tech. This was in August or September of '44. I was 17 and they shipped me out to Memphis, where I went through boot camp. I was to be a radio gunner firing one of these little 30-caliber peashooters out the back end of a torpedo bomber, the kind that George Bush, Sr. flew.

Booch: Right, right.

Humphrey: I was supposed to fire this little machine gun. So I got trained as a radio man taking Morse code. I turned out to get the top score in taking Morse code, so I guess dyslexia helped. I found that I could take five letter code groups at over 20 words a minute. And that's was pretty fast writing It wasn't typing. You had to write it out in longhand. I found that I would be three or four code groups behind in my writing and I could still write them all. It wasn't going through my brain at all. It was, sort of, wired through me and I was, sort of, hearing it and writing it and I had to keep my mind open to let it work. It was just extraordinary to me that your system could do that. So I concluded that the human system is capable of extraordinary stuff. We just don't know it. And that was amazing to me. I was valedictorian of my radioman class. Then they gave us a test. They wanted pilots. I first went to machine gun school where I was learning to fire a machine gun from a turret. Then I was sent to the navy V-5 to be a pilot. This started with V-5 training at college.

They sent me to a bunch of different colleges and then the war ended and I got a choice of signing up for five years and being a pilot or getting out. They offered us the option to enter the standby reserve. They said it was a no risk deal. I said, “I don’t want anything more to do with it. Good bye.” So I got completely out. My older brother also got completely out of the army Air Force. We then convinced my youngest brother to join the army. He was having all kinds of trouble at MIT where he had started at 16. He was way too young, but he completed one year. We convinced him to enlist in the Army of Occupation to go to Japan which he did. So he was in for only two years and he got the GI Bill. We all missed Korea and Vietnam. Which was extraordinary luck.

While I was in the navy in Memphis learning to be a radio man, the atomic bombs were dropped on Japan. I read everything I could in the navy base library. They didn’t have much but they had some books on astronomy and a few on physics, I read all of them. Then I decided that I really wanted to be a physicist instead of an aeronautical engineer.

College Years

When I got out of the navy, my dad, who by then was a colonel, was back from England. He was in Washington briefly with his wife and I was sent to Anacostia in Washington to get mustered out of the navy. He then went to the air war college where he was a teacher. This was in Montgomery, Alabama. He convinced me to go to college at Auburn, Alabama which I did. I could have gone to MIT but I would have lost three semesters of college credits and I was in a hurry, which was a dumb move. At Auburn I got into wrestling.

Booch: Before we go onto the wrestling bit I want to ask a question about, sort of, the state of the world back then. I mean you learned radios, machine gunning. I’m having a really hard time reconciling an image of you behind a machine gun but I’m trying to get it. But what was your exposure to anything in the computing field at that time because there wasn’t a heck of a lot going on, of course, but had you had any inklings of it? I mean, clearly you had a very fertile, inquisitive mind but did it ever lead you to the computing side of things? Were you cognizant of it?

Humphrey: This was ’46 and ’47. We didn’t have any computers then. I mean, there’d been early work by then but there was nothing. So this is really way before computers showed up and no one was talking about them or hearing about them. At least I didn’t hear anything about them. Nuclear physics was the top technology of interest and that was what I was interested in. I took engineering physics at Auburn and as part of that course I ended up taking a course in machine shop. I learned to cut metal, make screw threads, do welding, and other stuff. It was both fascinating and fun.

I loved to work with my hands and to build stuff and I’m not sure why I thought physics would have been great, but I did. I didn’t mention a high school teacher named Johnny Yarnelle. He was a whiz at math and science. He was trained as an English teacher but he ended up teaching more math and science and all sorts of other stuff. He also ran the Glee Club, wrote plays, and ran the school band. He was a fine musician. He was just an extraordinary man. When I graduated from high school, he also left. He was at the Forman School in Litchfield, CT, when I got there in 1935 and stayed until I graduated in ’44. He went to the University of Chicago to take a PhD in math. His undergraduate work had all been in English but he later told me that he so enjoyed teaching math to me and my brothers that he wanted to become a math PhD and he went to Chicago. He convinced me to go Chicago in physics, and he got me admitted to

graduate school in Chicago at the end of my sophomore year at Auburn. Chicago did crazy things like that at the time.

Booch: Isn't that where one of the first nuclear piles was done under the stadium at Chicago or am I thinking of a different place here?

Humphrey: That's the place and Enrico Fermi was one of the professors. As a matter of fact I had Fermi as my professor for nuclear physics.

Booch: Well, you can't get a better professor than that around that time, wow.

Humphrey: He taught me something though that changed my life. He taught me I wasn't cut out to be a theoretical physicist.

Booch: And how did he teach you that?

Humphrey: Well, I just listened to these lectures. Remember, I just had a sophomore background. All of these other graduate students had all finished college and so I'd been in a hurry and unfortunately this time I was in way too big a hurry. I discovered that just plain hard work didn't do it. And so, after about a year and a half, we had to take qualifying exams to see who could stay and do graduate work for a doctorate. I spent about a month and a half in at my father's apartment in North Chicago studying. Father had by then moved to Chicago and was running the local CIA office.

Booch: Oh, wow. There's a story in there but let's-- wow, I'll bookmark that one.

Humphrey: So I went and lived with him and studied-- went through all the physics basics in real detail. I'd had a lot of physics courses by then and I really studied it up in preparation for the exam. There must have been about 60 to 70 taking the exam and I didn't know it but they had decided that they would only pass 12. I took the exam and actually finished it first. I thought I did pretty well and I must have done so because I came in 13th. While this was a failure, but it was actually pretty damn good. These were all college graduates who had taken four years of physics and I'd only had two, but I still came in thirteenth, but I was in too big a hurry. I didn't check everything properly. I was just too cocky, I guess.

By that time I was working in the nuclear physics lab running the betatron at night. When I failed the exam, they gave me a bachelor's degree in March of 1949. It is hard to believe, but that was over sixty years ago now. I then went on and got a master's in physics at IIT [Illinois Institute of Technology] and discovered that physics wasn't what I wanted to do. I was still working full time trying to figure out what I wanted to do and I had some friends who were going to business school. So I decided to try that. It sounded like something I could get through pretty quickly. I got a master's in business in 12 months while I was working. I was working full time at the same time.

In the spring I wanted to register for five courses. I had taken four courses all along, when three were considered a full load. But I needed five courses to graduate, so I wanted to register for five. They wouldn't let me without the permission of the dean. So I talked to the dean. I told him I needed to register for five courses so I could get my master's degree in the spring. And when he looked at me he said, "Aren't you working?" And I said, "Yeah," and he said, "How much do you work?" I said, "Full time." "And

you've been taking four courses all along?" And I said, "Yeah." He said, "You're crazy. Okay." So he let me take five courses. But fortunately I could do a lot of work while I was running the betatron at night. I got an MBA and, even though the courses were pretty easy after physics, I learned a lot. My major was in manufacturing and the professor Judson Neff said, "The three most important things about manufacturing are planning, planning and planning." He drove that in. It was only a one semester course, but it taught me a lot. I also took cost accounting which was an amazing revelation. It was fascinating and interesting. By then, I wanted to get back to technical work, so I continued at night school taking courses at IIT in electrical engineering. When I got my MBA the university asked me to become a director of scientific personnel with a new lab they were starting. So I took that job. It was a very nice job and I enjoyed it. I had a secretary. My first job out of college I had a secretary.

Booch: Wow, nice.

My First Job

Humphrey: This secretary taught me a lot. She was an executive secretary whose husband was at the university to get a PhD. Her name was Gloria Gentilly and she taught me everything about running an office. I was director of scientific personnel and I also worked as an engineer. I also ran the security shop and got everybody cleared. I also had to be cleared for a Q clearance and Top Secret. One day I was sitting in my office and a guy from the FBI wanted to see me. He came in and started to interview me. He said, "I want to ask you a bunch of questions about one of your people we're getting cleared for Top Secret and Q clearances. As He started down his list of questions and he said, "What do you know about this guy Watts Humphrey?" I said, "I actually know quite a bit." I gave him my card and he turned pale, but I got the clearance.

I was the director of security so I went down town and talked to the Air Force colonel about how to handle the most likely security problems. I got to know him pretty well. One day I got a call from the director of the lab, a Dr. Hoagness and he said "We have got to get Top Secret clearances for," and he gave me a list of people, "by tomorrow." One of them was Enrico Fermi, who had been a member of the fascist party in Italy. This would not please the people who granted clearances. But they all had AEC (Atomic Energy Commission) Q clearances.

So I called my colonel friend and asked him, "How do I do this?" He said, "Well, if you can actually get me proof that they have Q clearances, then I can issue the Top Secret clearances right away." I said, "Great. How do I do that?" He said, "You have got to get it from the AEC." So I said, "Okay," so let me find out. So I called the AEC and everybody kept referring me around. I wouldn't let go until they gave me somebody else and I finally got to the records clerk down in the basement somewhere with the AEC. He was very nice. I said, "Here's what I need. You don't send it to me. I want it sent to this Air Force person." I gave him the colonel's address and phone and he got all the records and he wired the information to the Air Force colonel. I had the top secret clearances the next morning. About two months later there was a new AEC regulation about "no one will ever call the records clerk in the basement of the AEC." I had found the crack in the dyke. That was my greatest achievement as a security officer.

Booch: So a question for you along the way. In your comings and goings in that community did you ever run across Feynman by any chance?

Humphrey: I did not. No, I did not. And I knew Teller, Fermi and Urey. Well, they didn't know me but I knew who they were and so it was an exciting period. I remember sitting in a meeting where a graduate student was giving a talk on something or other and it was a very pleasant discussion and the door opened and a guy walked in and sat down next to me and everybody just stopped. It was Harold Urey, the Nobel prize winning chemist, and we all knew who he was. Anyway, we all sat there stunned. He said, "Please, go ahead. I'm here to learn, so you just go ahead," and he asked a few questions. He was just a marvelous gentleman and not particularly self-assuming or anything. So it was very impressive but it was that kind of environment where you could get to know these people. And so I was working at this lab and at the same time I was taking graduate courses in electrical engineering at IIT. That's when I really got excited about computers. My principal technical work at the lab as engineer was looking into analog computers. And after a couple of years there I realized I wasn't going anywhere. The lab wasn't producing anything that was useful, even though it had Fermi, Urey, Teller, and all these people on its staff. I also knew Richard Garwin. I don't know if you've heard who Richard Garwin is.

Booch: I do not know.

Humphrey: You've heard the arguments about who invented the hydrogen bomb?

Booch: Right, right.

Humphrey: Well, Teller was working with our lab and, as part of that job he was down in Los Alamos trying to get the scientists to design a hydrogen bomb. They literally couldn't do it. They knew theoretically what to do but they couldn't produce a design.

Booch: He had quite a passion for that in reading some the history… that was his obsession if I recall.

Humphrey: He had a vision of how it ought to work. The physics was pretty well known. In the spring--I've forgotten which year - he got hold of Dick Garwin and said, "I want you to go down to Los Alamos for the summer." Dick had been a graduate student with me and he'd gotten his PhD about the same time I graduated. And a nice guy, I mean, I knew him quite well and he went down to Los Alamos for the summer, he designed the hydrogen bomb.

Booch: Oh, my. "What I did on my summer vacation?"

Humphrey: Yeah, and then he came back and it worked and he just went on back and was a professor at Chicago and then he later moved and was a researcher at IBM, where I knew him. A few years ago there was this big discussion with Teller in Congress about who invented the hydrogen bomb. All these guys from Los Alamos were saying they did it. The Senators asked Teller who did it and he said it was Dick Garwin did it. So the press interviewed Dick, who was at IBM at the time, and they asked him, "How come you never said anything about this?" He said, "I decided that you could either make great contributions or try to get credit for them, but you can't do both," I thought was an amazing line and I've tried to remember ever since. He just got satisfaction from doing great work and wasn't particularly interested in getting pats on the back and credits. That was just extraordinary.

Wrestling

Booch: Yes, what a humble guy. Hey, I want to go back to two things, before we get much further in the genealogy and I'll lay them out. Tell me about your wrestling career and passion, because I understand you had a coach that was in some way associated with the Olympics. And you used the phrase that, at night you'd go in and run the betatron I think it was, and I'm really curious, what does that mean? I have a picture of this, you know, old B-type science fiction movie surrounded by dials and flashing lights.

Humphrey: Well, let me start with the coach, before I forget it and then I'll come back to the betatron.

Booch: Sure.

Humphrey: The coach was called "Swede" Umbach. When I got to Auburn, my dad convinced me to try wrestling. I'd never done it, but in the Navy, I wanted to wrestle instead of box. They showed us a little about boxing and wrestling, but I didn't know much about wrestling. I remember that we wrestlers would always take the boxers. I decided to go out for the wrestling team at Auburn. There were two of us trying out for light heavyweight. That was the 175 pound weight and the next was heavyweight, unlimited. People were a lot lighter then.

We were all new. None of us had ever wrestled before. And the coach arrives and he'd been coaching in Oklahoma, and where they had extraordinary wrestling teams. He was about 155 pounds, but he could take any one of us. The first thing we did every day, when he showed up, we'd go out and run the track up to a half mile and he'd be right in front. We'd all do it together, and then we'd come in and wrestle. He'd wrestle us hard, it was really tough practice, and he got us in great shape. He taught us the basic holds very quickly and then we had our first meet. It was out of town, it was up in Tennessee.

I had made the varsity team and I wrestled this guy who was my weight. He was exactly the same size but was more experienced. He'd wrestled for two or three years and he was very good. He couldn't pin me, I couldn't pin him. After the standard three rounds, I was absolutely beat. We were tied and had to go into overtime rounds. We were exactly tied on points and time. That's how they did the points, what moves you got and how long you were able to be on top of the other guy. We were exactly tied after three, three minute rounds, and I was just laying there, flat on my back, seeing black, I couldn't see anything, and the coach was whispering in my ear. He said, "He's more beat than you are." He said, "When you get out there--" I would start on the bottom-- he said, "The minute the ref blows that whistle, you explode, just get out of there, he won't be able to hold you." And so I did and I don't know where I got the energy, but I did. I got the escape, which was worth a point. And then it was my turn to be on top of him and by that time, he was absolutely exhausted and neither of us could do much of anything, but I stayed on top of him and got the point, so I won.

Booch: Marvelous.

Humphrey: One of the guys later, one of our guys who had been on the timer table, came to me that evening and privately told me, "I made a mistake and you really didn't have the time I gave you." I said, "So I really lost?" And he said, "Yep."

Booch: Oh, no.

Humphrey: Well, it was too late. I mean it was all over and it was done I don't think he even told the coach or anybody else, but I knew that this guy had been able to beat me. I think I was undefeated for the rest of the season. The heavyweight was also undefeated as were several of the others. We then ended up in a final AAU tournament for the Southeastern U.S.

By the way, on campus, I was called "Slide Rule" because then we didn't have calculators, we had slide rules on our belts. Every other team in the school was failing and we were winning. So for a wrestling meet in the gymnasium, the place was absolutely packed. Wrestling in front of a cheering crowd of fans, with a team that is motivated and working well together, and with a hard-driving and motivating coach, was the greatest experience you could ever imagine. It was just marvelous. The teamwork, the coaching, the enthusiasm of the crowds, and all of that helped to produce an absolutely extraordinary performance. People really put themselves out, under those conditions. You're doing it not just for yourself, but for your team. It was just an amazing feeling. In the finals of the AAU, which is 13 southeastern states, guess who I ran into?

Booch: The same guy you wrestled at the very beginning story.

Humphrey: That's right, same guy and I were in the finals. And so we came out and started off and he was a push over, I couldn't believe it. I had beaten him and I think it made an enormous difference to him. It didn't-- I mean, I knew he'd beaten me and it didn't really affect me, because I'd won everything since and I'd been a winner and he wasn't a winner. And it's amazing how that affected his performance. I was really quite surprised, but a very nice guy. So we had a marvelous team. After that year, I got transferred up to Chicago. I went out for wrestling in Chicago and the coach there was a nothing. He didn't push anybody and nobody worked very hard. I did extremely well and I kept working at it myself. And one day the coach came over and said, "Hey, there's a guy here that's your weight who would like to work out." And I said, "Oh, okay." So he came over and introduced me to him and said, "Okay, can we do it tomorrow?" And I said, "Sure." So the next day, we picked a time, suited up and I came out there and we started working on the mat. And he could do anything with me. I could have been nine years old. I couldn't believe this guy. Here I was winning all these tournaments and I was Southeastern States Light Heavyweight Champion. I'd even wrestled heavyweight once. I had done so well that I just was full of myself. I'd been undefeated and this guy could do anything with me. Pin me, or whatever. It turned out that he was on the U.S. Olympic team. I realized right then that world-class performance is something else altogether. I mean this guy was really extraordinary. So whenever I feel that I am pretty good, I remember this guy, That calms me down real quick.

Running the Betatron

Humphrey: Okay, now let’s go back to the other part of your question about running the betatron.

Booch: Yes, what does that exactly mean?

Humphrey: Well, a betatron is a big nuclear accelerator and the one they had there was 500 MeV, a half billion electron volts of energy. The big cyclotron was next to it. A cyclotron accelerates protons and heavier particles, but the betatron just accelerates electrons. The electrons were in a large vacuum tube doughnut that was about 50 inches across and 6 to 8 inches in diameter. The electrons are accelerated up to very close to the speed of light, and then they hit a target.

The betatron is basically a resonant magnet at 60 cycles. The enormous magnet could resonated with a great big condenser bank that filled a large room – about 20 by 40 feet and about 8 feet high. The thing actually oscillated at 60 cycles (now Hertz) at about 20,000 volts or more. And so our job was to run this thing for the scientists when they were doing experiments, Fermi and a whole bunch of people did experiments there, and they'd leave the experiments and we'd run them at night. And we all had badges-- our physical records were all stamped RA which meant radioactive.

We would periodically tape our badges in the betatron beam, and we never got a report on them. We'd just wear them, nobody ever looked at them, because we pumped enough radiation through most of our badges to kill an army and no one paid any attention, never heard a peep. So I-- when you've got a bureaucracy, you ought to check it occasionally. Some of the scientists, when they'd run experiments, they'd come in at night. And there was a lady, Leona Marshall, who would walk around the experiment right by the radiation. We had all these concrete blocks with lead instead of gravel, stacked around to protect us. Someone could go in there, however, and we could watch it. We'd go upstairs to the balcony and you could look down and see Leona as she walked around, ducking when she walked by the beam.

Booch: Oh, my.

Humphrey: I mean, I couldn't believe this lady. But that's what she did. I learned a lot about electronics from that job. We actually blew a power company substation once because we wanted to soup up the power enough for the experiments. One of my side jobs was to build a bunch of spark gaps, or controlled arcs, that would dump the condenser bank in a controlled way.

Booch: Right, right.

Humphrey: To make the spark gaps, I had to machine graphite and lead which was tricky. I was doing all of this at night, while I was taking my business school courses.

Booch: Wow, quite a Renaissance man you were. So, thank you. I imagine that you were-- the room was filled by the sound of the 60 cycle hum as well, too.

Humphrey: Oh, yeah, lots going on.

Booch: Sure. This is utterly fascinating by the way. I would imagine this is not the normal thing that comes up in the interviews you've had so far, but I'm loving these human stories. So let's bring this up then to where we had been, right before those two questions. You were starting to see computers around you. You had, you know, been involved in some of the analog computing, saw that kind of dissipating. What was the first computer that you had your hands on?

Humphrey: Well, let me move back to the story a little bit, how I got there, if you don't mind.

Booch: Sure.

Sylvania

Humphrey: I decided to leave Chicago and get a real job in industry. I'd been taking graduate courses at IIT and was debating getting a PhD. The professor gave me a really nutty thesis topic, which was way beyond me. Wanted me to come up with a theoretical analysis of plasma oscillations. That was not something I was interested in doing. I took a shot at it but got nowhere. So I decided to look for a job in industry and sent inquiries to RCA and IBM. My younger brother was then working with Sylvania up in Boston, so I decided to talk to them too. I had a car at this point and drove back for interviews at RCA, IBM, and a couple of other places, as well as at Sylvania. Sylvania gave me a marvelous offer so I decided to go there. My younger brother and I shared an apartment, which was great.

Shortly after I got to Sylvania, they sent me to a summer course on computers, This was my first year there in 1953, The two week course was on the Whirlwind computer. It was taught by a couple of guys from Cambridge, Cambridge, England. Oh, Lord, I can't remember their names. One of them just won the IEEE award. He and Parnas tied for.

Mrs. Humphrey: Maurice? First name is Maurice.

Humphrey: Morris-- is it Wilkes?

Booch: Oh, Maurice Wilkes.

Humphrey: Yeah, Maurice Wilkes. He was one of the professors teaching the course. So I walked into the course and there registering everybody was this pretty young lady, and so I spent my two weeks trying to figure out how I was going to meet this lady. Well, I did chat with her that day, during a break, and it turned out she was from Chicago and so we had mutual friends. I took her to lunch and my major for the course was Barbara. We've now been married for 55 years, with 7 children and 11 grandchildren, so I think it'll last.

Booch: I think it will stick. And was that her voice I heard in the background a moment ago?

Humphrey: What?

Booch: Was that her voice I heard in the background a moment ago?

Humphrey: That's right, telling me Wilkes.

Booch: Well, congratulations. Wow, like 55 years you say, that's remarkable. Congratulations.

Humphrey: Yeah, I learned an awful lot in that course. I wrote a program for the Whirlwind and I got totally committed to computers and to Barbara. She had been a math minor and English major, and they wanted somebody who could help with writing and presentations. The scientists were close to illiterate, but not quite. She was helping them. But she got me a copy of-- was it Herman Goldstein and John von Neumann, if you've ever heard of it. Anyway, it was a book that they wrote on planning and programming computers.

When I started at Sylvania, I'd had all these theoretical courses on electronics, electrical engineering, and physics. Since I had this tremendous background, they gave me a job as a manager. So, starting with my first job out of college, I was a manager. I never did work as a worker. They put me in charge of the circuit design group for this big cryptographic computing system. It wasn't a computer, but it had lots of computer logic. It was a digital crypto system with 5,000 subminiature vacuum tubes, it was an enormous system. Building a highly reliable system with 5,000 vacuum tubes was a challenge. This had to be reliable under military field conditions. So it had to be in ruggedized racks, work under extreme humidity and temperature conditions, withstand shock and vibration, and handle wide power fluctuations. Designing it was an enormous challenge. The problem for me, of course, was I knew nothing about designing circuits and I had this group of circuit designers working for me.

I decided, instead of trying to tell these guys what to do—since I didn't know what they should do-- I spent my time asking questions. I was quite frank with everyone. If I didn't know something, I'd say, "Okay, why does that work that way?" And so I treated my job as a learning experience. What surprised me was how much the engineers loved it. They loved to talk about their work, they loved to explain what they were doing, and they didn't look down on me for a minute, even when they knew something I didn't. It didn't bother them, they were perfectly happy. They were proud that they could explain something to me. I learned more, in that brief period than I ever did in college. And the team was just extraordinary. We got patents on some circuits and produced some great products. I learned a lot real quickly and was soon able to start helping the engineers when they had design problems.

I have followed this same management style ever since. It turns out that, when you manage thousands of people, you're going to have lots of people that know more than you do. With this style, you can work with people who know more than you do. If you can't do that, your future's terribly limited. I was enormously fortunate to learn that on my first real job in industry.

We had a great team. A few months after I got to Sylvania, my boss, the program manager, got promoted up to another job in the laboratory. It was actually to work on the countermeasures system for Hustler B-1 bomber. So they made me the program manager for the crypto system. I had the electrical engineers and mechanical engineers working for me. The minute I got promoted, the mechanical engineers came to me. They'd been asking for a couple of months, about the power requirements.

So when I became program manager, they came back to me and said, "We've got to get the power requirements." I said, "We haven't designed the circuits yet, how can we give you the power requirements?" They said, "Well, if we're going to meet the schedule, we've got to get all the structural frames ordered and, to do that, we've got to order the transformers." So I said, "Okay, well, how long does that take?" Well, it turns out that if we didn't get the transformer order in next week, we couldn't make schedule. the transformer was a long-lead item. I said, "Oops." So we made a power estimate right then. We were very cautious and put a fair amount of fudge on it. We actually had enough, which was fortunate. At the same time, I said "Let's make a detailed schedule for the whole program."

We got with the manufacturing guys and laid out the plans and put the whole schedule together. We found that we barely had time to get the job done. Later, in the middle of testing the hardware, the circuits were working and everything was great. We were preparing for the final testing and for the Signal Corps to come up and review the system in the next week or so. I was home one weekend and I got a call from the guards, there was a hurricane coming. In those days we didn't have hurricane warnings, and we were in a basement lab in downtown Boston. The guards called me and said, "Hey, there's water coming up

through the floor." We had all this equipment on live test. It was all sitting on the floor. So I raced into the lab real quick and, and I was amazed to see the entire team show up, without my even calling them. They all came in, we killed the power, and we got it all jacked up and off the floor. I mean, the team was-- it was just astounding. So they showed up without even being called and they saved the day for us.

Booch: Wow, that's remarkable.

Humphrey: Isn't it. Amazing.

Booch: So you were doing cryptanalysis work around then. I'm curious if you had any connections with Turing or had heard of the Colossus around that time? Was that on your radar?

Northeastern University

Humphrey: No, that wasn't, but when I got to Sylvania and I started working on this machine, I got very interested in computers. So I went over to Northeastern University. It was right there because our lab was on the campus. I asked if they had any night school courses on computers that I could take. And they said they didn't but they wanted to know my background. When I told them, they talked me into teaching a course on computers.

Booch: Okay.

Humphrey: I was quite used to teaching because the professor I had at IIT, George Cohen, was wonderful. I took every course he taught in electrical engineering. On occasion, he would call me up and say, "Hey, look, I can't be there tonight, will you give the next lecture?" And so I generally arrived at class prepared to teach. I was working at the Chicago Midway lab in the day and taking these courses at night. I was used to teaching, even stuff that I didn't know. Since I had been able to do that, I decided that I could probably teach a course on computers.

I went over to the MIT library and the Harvard library. I had previously talked to Howard Aiken about taking a PhD at Harvard but he wanted me to take a lot of formal math that wasn't the least bit interesting. So in preparing to teach this course at Northeastern on computer design, I read the von Neumann book and I went to the libraries of Harvard and MIT and read everything written on computers. And there wasn't a whole lot. It is amazing that I managed to fall into the computer field at a time when you could read all the literature on a subject. Can you imagine doing that today?

Booch: It'd be tough.

Humphrey: Yeah. So I read everything about it, everything they had. I got the Kiester, Ritchie and Washburn book by the Bell Labs folks, which was about switching design. It was a very good book, but it wasn't about computers. It was designing, essentially, telephone switching exchanges. It had a lot of useful stuff in it. I put this two-semester course together, and the first class started on the 13th of September, in 1954. I had 13 students, and that day we moved into a house in Cochicuate, MA at Number 13 Bald Rock Road. So a lot of 13s all hit on that one day.

Booch: Marvelous. Those 13 students, have you had any further contact with them? Do you know where they've ended up?

Humphrey: I have not. They-- most of them worked for Honeywell and they were all designing the Datamatic computer. They probably knew more about computers than I did. But I followed the same kind of teaching strategy that I had used in management. I would have them explain stuff to me. They knew a lot that I didn't, and so we had a marvelous class. I learned a lot from that course. I then went through my notes and I put them together into a framework of the course the next year.

I had very detailed lecture notes and, at the end of the year, I decided to see if I could turn it into a book. So I contacted McGraw-Hill, to see if they'd be interested, and they were. So I wrote my first book. The second year I taught the course, it wasn't as much fun because I knew so much more at this point than I had the year before, and the students treated me more like a professor than a colleague. I had notes for my manuscript, which I was teaching from now, and the third year I taught, I had the published textbook. I taught that for four years.

Booch: What was the title of that book, do you recall?

Humphrey: *Switching Circuits with Computer Applications.*

Booch: So mostly what you taught-- I would imagine that's out of print these days, I don't know.

Humphrey: It's out of print, but it was used as the text at MIT and Purdue and some places in South America- it sold about 20,000 or 30,000 copies.

Booch: In fact, thanks to the power of Google, I see that one can still buy some copies, so it's available.

Humphrey: Wow, it's still available.

Booch: So most of your course was, therefore-- was certainly not programming it sounds like, but more on the machine side, the architecture of the computer. Would that be a fair characterization?

Humphrey: I had not done any programming, except what I'd done at that MIT course.

Booch: Got it.

Humphrey: And so I wasn't really a programmer at that point.

Booch: In your comings and goings, did you run across von Neumann or Eckert and Mauchly as well, too? Where were they in the midst of this?

Humphrey: I didn't run into any of them. In fact, the people that I ran into at the Barta Building, a lot of very bright people, they all disappeared from the scene, never heard from them again, except for Wilkes of course and those guys. I ran into Wilkes, by the way, later. He and I both were giving keynote

speeches at a conference up in Canada, so I got to chat with him there again. He didn't remember me, but I certainly did him. But yeah, so I learned a lot. The lab moved at this point out to [Route] 128, to a new location, and I remember we moved our big 5,000 tube bunch of stuff, great big relay racks and we had another system that was there. It was a 48 channel PCM communication system with 48-channel voice-- it was a bit stream that they had actually multiplexed 48 channels. And so we got this analog stream that was converted to digital and it was a 1.17 megacycles, I believe, now Megahertz, and that's what we had to encrypt. And so we got the system all set up. Now remember we're in a new lab, we got this whole bay of stuff all set up and they told me to power it up.

So I started to power it up. I didn't get very many of the units up before it blew a fuse and so all their guys came over to see what was going on and the building maintenance guy, who was running all of that stuff, he said, "Well, we'll put in a little bit bigger fuse." And so they put in a little bit bigger fuse and so I went a little bit farther this time and blew that out. And the electrical engineers who were putting this stuff in, they came in and they kind of laughed and they said, "He said to put in a bigger fuse so we did." And I said, "Oops." And he said, "Okay fire it up." So I fired it up, and I got most of it up, not all of it, but most of it up and it sounded like a bolt of lightning. I mean, when it blew that fuse, it really-- it blew the fuse box. It got-- the electrical guys came over and said, "Come over and see this, you've got to see this." So I went over with them. The whole wall was black. It had taken out the whole fuse box. I mean, this was a lot of power that I was drawing with this thing. So in any event, they got it all fixed ultimately, it took them a little while. And so we got it all back up and I had been trying to figure out I was-- how we were going to test all this. This had been somewhat earlier. We were due to get a demo for the Signal Corps and for some other gentlemen, who arrived from an unnamed agency. And the question is, when you've got this random digital data stream, how do you determine that it's working. I mean, the circuits were working, but do you know if the logic works?

So I worked through the logic of this and figured out how I could get it by connecting the input to the output of the crypt box, how I could get it actually to fall into a pattern and so I did. I was able to set up-- worked out a bunch of digital standing-wave patterns and we used them for testing the system and, when the Navy-- when the Signal Corps came to get a demo, I probably described this system for getting this to fall into these synchronized patterns, to self-synch, and demonstrated, the Signal Corps guys were all excited. And these three men, in black suits, didn't say a word. And I realized later I had killed the system right then. I mean the crypto guys weren't the least bit interested in a system you could get to synchronize like that. It would have been tough to test if I hadn't. But in any event, so the first system, we actually built it, shipped it off, and it was installed, apparently, in the Pentagon, linked to some unnamed location. I learned later than it ran for 5,000 hours unattended without an error.

Booch: Wow.

Humphrey: For 5,000 tubes, that's pretty good.

Booch: That's pretty amazing.

Humphrey: Yeah. So those circuit design kids-- they were all young, they did an extraordinary job. We had it derated so you could bring the power down 50 percent and you could do all kinds of stuff with it. It would run from minus 30F to plus 130F in high humidity and it was an amazing system and it just worked forever. So it was a tragedy it didn't continue to get built, but it didn't, but anyway. And at that time, I was running this group now, I had several projects at the Waltham Lab at Sylvania but we were running out of

work. And they had work for the B-1 bomber and they also got a contract for the BMEWS project, you know, the radars across the northern tier, so it was another group that was doing this. The guy who'd been my previous boss and been promoted, he was up in that group.

Building a Computer Group

Humphrey: I was trying to build a computer group now.

Booch: Now, what year would this have been?

Humphrey: This would have been in 1956, I think it was.

Booch: Wow, I didn't know the B-1 was in concept way back then. I mean, it takes decades for those kinds of things.

Humphrey: The B-1 was-- they were building-- it was, you know, anti-- it was electronic warfare.

Booch: Right.

Humphrey: What do you call it? Yeah, it was jamming stuff and everything else, for the B-1 bomber. It was actually being built then. My brother-- that's what my brother was working on. He'd tell me these wonderful stories. I remember that he this stuff and he was telling me about it, and they were testing it and these people coming in for a demo, the management team, not outsiders, and they rigged up a little tube that would run through the machine and he would take a big puff of cigarette smoke and puff it through this tube and so you get a puff of smoke ends up coming out of the top of the machine and he said management went wild.

Yeah, they shipped the first B-1 electronic countermeasures, that's what it was ECM; they shipped the first unit down to Texas from Boston without wrapping it up. And when it got there, it'd gone through all kinds of stuff, and it got there and it was just full of bugs. They called it a very buggy system. But it had traveling wave tubes and all kinds of stuff in it and that's been retrofitted, but that was back in the mid '50s, early to mid '50s. He went to that work in '53. So in any event-- but that group had gotten a big hunk of the BMEWS contract and it included a bunch of computers and all kinds of stuff and so they were-- they wanted to get my group to go work with them, they wanted all our people. Are you still there, can you hear me all right?

Booch: Yep, I hear you just fine.

Humphrey: Okay. And so they-- the big move with management was to disband our group and just distribute our people among the countermeasures crowd and the BMEWS crowd and I wasn't interested in that because I wanted to work on computers. So one day a guy came to see me, a marketing guy, and he said, "Hey, we got this bid I heard about, that the Navy and the Training Devices Center down in Port Washington has put out for what they call a UDOFT computer." He said, "Are you interested in that?" I

said, "Yeah." Can you make them a bid now, it's due back in just a couple of weeks. I said, "Oh, okay, well, let's see what we can do."

So he and I raced down to Port Washington and talked to the Navy and they finally agreed to put us on the bid list, but there were a bunch of other people on that bid list, about two or three other companies. They weren't the big guys, but they were some important companies. So in any event, we got the bid stuff, put together the proposal, and got it in in time. We also, at the same time, like a day later, the guy came in and said, "Well, there's an Army bid from the Signal Corps for a computer called MOBIDIC. Are you interested in that?" I said, "You bet." So we raced down there and we talked to them and they put us on the list that had also already been out, but that one had bidders like IBM and RCA, and Philco, and there were about four to six different bidders on that. And so we bid both of them and, of course, we were facing getting closed down as a department. We convinced management to not close us, until we heard about the bids so they agreed, and we won them both.

Booch: Oh, my goodness.

Humphrey: We won the MOBIDIC and UDOFT contracts. As I say, the other bidders included RCA and IBM and Philco, and so I was kind of staggered that we won this. And all of a sudden, here's this little group, we had two big computers to build, so we did do a mite of recruiting pretty quick and we got a good team together and they did a marvelous job. Delivered them both, got them both working, we had them up on the floor. So I learned a lot from computers. That's why I didn't learn to program because I was the architect on both machines and the architecture of the UDOFT was already designed. I mean, the University of Pennsylvania basically designed it and our job was to build it, so we-- they told us how to do it and everything. I mean, we had to come up with the circuit designs and get it built, but we didn't have to do any programming on it. We also didn't have to do the MOBIDIC programming either, as a matter of fact, but we needed to design the instruction set.

Booch: Any surviving copies of those machines?

Humphrey: The MOBIDIC, I think there is one. I'm not sure where it is. I've been down to-- I've given talks and a bunch of papers on it. There's not much on UDOFT. I don't know what happened to UDOFT, I really don't know, but I did get to fly a jet aircraft, but it was in a simulated aircraft on the UDOFT. That was a rather funny experience. I was-- I wanted to get in and fly this thing. It was an F-104. They had two airplanes. That was the real test of it, we had to build this machine-- it was a pretty powerful machine. It had a 10 microsecond add time and a 20 microsecond multiply time, which was pretty damn good in those days. It had separate memories for instruction and programs-- no, instructions and data.

Booch: How much memory would these things have had?

Humphrey: I think the UDOFT, it was like 10,000 to 20,000 words, it wasn't big. It had, I think, a 20 bit word. MOBIDIC had a 38 bit word, and I'm trying to remember how big the memory was, but MOBIDIC had to be designed to be an expandable system. And so it had to be able to grow, in terms of memory size and stuff, but nothing else, memories and channels that could grow, but <inaudible>. But there were a bunch of other versions of it, which we got-- some of which we got, but IBM and RCA and Philco and others were all still trying to get some of the fallout of this. It became an enormous program, lots of people working on it. And it was doing great, and we built several machines for Europe. I believe there is one MOBIDIC down in Monmouth, I'm not sure. But we had several of them in Europe. They worked fine. The

whole program got killed, though, with Viet Nam funding, so that kind of torpedoed that. But in any event, I learned a hell of a lot about computers and my programming experience was quite different from most, because when I would write programs, if I didn't like the way things worked, I'd change the instruction set.

Booch: Very good. And, of course, back then, programming was still at the assembly language level primarily, was it not?

Humphrey: Right. We were writing assembly language programs and all that sort of thing. One of the computers we got out of this, the fallout of the MOBIDIC project, and there were really two kinds of fallout from that. One was a program-- a machine that had to have multiple computers running off multiple memories with multiple channels and it's-- because it was supposed to be able to fail soft so you'd be able to have one computer and we didn't have to program it. We had to figure out how to build it and that turned out to be a bit of a challenge but we did it and we could do it, so we had real-time connection-dependent computers would sense and check each other and all that sort of thing. So we actually built machines like that, designed them. They built them after I left. We designed them in 1958, '57, this sort of timeframe but we learned a hell of a lot about how to do all kinds of stuff then.

Booch: And would it be fair to characterize it, most of these machines are really not general-purpose but they were specific purpose things then?

Humphrey: Well, the MOBIDIC was a very general-purpose machine. It was built basically like an IBM 704. It was much more powerful than the 704 and it was the most powerful machine available at the time. It was transistorized. It was I'd say the first large scale general-purpose transistorized machine. It was built with unique components, though, unique transistors and I had a very good crew working for me. I hired a PhD who ran our circuit design work and he was very good. We had a whole bunch of folks. It was a marvelous group.

Booch: Were you guys manufacturing your own transistors? I mean, even those things were not common beasts back then.

Humphrey: No, no. We would buy transistors and we found that commercial ones were available that were able to meet the military specs. The MOBODIC was in a trailer and it had to meet military conditions. We had the same problems that we had with the other one to build a very highly reliable computer that could run with shock and vibration and temperature and humidity, all that stuff. And so we had to build some very reliable circuitry and packaging and so it turned out to be an enormously powerful machine and very effective. So they were basically running without any problems. Those machines were just remarkably reliable.

So we were trying to get it used on BMEWS and trying to convince the Sylvania folks to start marketing MOBODIC. It was certainly a better machine and it was marvelously available. I remember I put this proposal together. My immediate management all backed it, had to get the backing of the lab director, however. So I went in to see the lab director who I will not name. I went through the proposal and he looked at me and he said, "How long will it take before this will pay off?"

And I had my MBA background and stuff and I said, "Well, it is going to be about 10 years before you get real money on this because it will take quite a bit of investment." Not a quick and easy thing and ten years

may have been optimistic but I thought we might be able to. We had the machine design. It did work and so when I said 10 years, he looked at me and said, “But I retire in five.” <laughter> So I said, “Okay, thank you.” I wrapped up my stuff, left and went out to call my dad and figure out how I apply for a job at IBM.

Applying to IBM

Booch: So what did you know of IBM at that time? They had been another bidder it sounds like and what was your impression of IBM?

Humphrey: Well, my dad had been, as I said, on Wall Street and he was the chief investment advisor and treasurer for what was called General Reinsurance then. I believe that became AIG but I’m not sure and he made a whole series of recommendations to the management back in the 1929, ’30 timeframe for the stocks they ought to buy, one of which was IBM. He said that’s the only one they didn’t accept. They bought all the others. And he said, “If they had bought that they would have been worth more than all the others put together.” <laughter> He didn’t buy IBM stock but he advised several good friends to do so. And so I had been buying IBM stock all along. I had been putting my time in Sylvania but every nickel I could scrape up I put it in IBM.

Booch: And thanks to Google I see that in December of 1960, IBM stock was selling about $20.00 a share.

Humphrey: Yeah, whatever it was it was the-- well, I’ll tell you. In 1953 it was pretty good. <laughter> So I joined Sylvania in ’53 and I started buying IBM stock right away whenever I could and my wife at one point, Barbara said, “Why are you spending your time at Sylvania and your money at IBM?” And so that got me thinking and so when this lab director said this, I said, “Well, maybe I ought to think about IBM.”

Booch: You married a very wise woman it sounds like.

Humphrey: She’s sitting right here, so you can say anything you want.

Booch: You’re a very wise woman.

Humphrey: Say hello to Grady, Barbara.

Barbara: Hi there.

Booch: Hey, greetings.

Humphrey: But in any event, so I got a hold of my dad and said, “I’m thinking of applying to IBM. What would you do?” And he said, “Well, I’ve got a friend who I think might be helpful. He knows the IBM

people pretty well, so let me send a letter. Can you send me a write up on yourself?" So I said, "Sure." So I sent it in. I'm trying to remember his name. I've gone blank but it was-- yeah, my dad said he was a wonderful guy. In any event I'll remember his name at some point here soon. But my dad sent him the resume.

Booch: Now, what year would this have been?

Humphrey: This was in early '59.

Booch: '59, okay. And IBM at that time according to my calculations had about 94,000 people in it, so it was a much smaller company back then.

The Watson Interview

Humphrey: Yeah, it was still pretty big. I mean, it was a lot bigger than Sylvania, I'll tell you. So I got a call that I was to come down to New York and see Tom Watson.

Booch: Tom Watson, Sr.

Humphrey: Junior.

Booch: Oh, junior? Okay.

Humphrey: This is '59. Senior died in '56. So this was Tom Watson, Jr. who was then president and CEO of IBM. I said, "Well, okay." So I didn't know what was going on, so I went down to New York and walked into 590 Madison Avenue where headquarters was at the time and they took me up to the 16th floor and waited there a few minutes and then I went in and there was Tom. Marvelous gentleman. We had a great chat and he was talking about things and who did I think was going to be the big competitor and I-- as we talked about it I told him I didn't think RCA was going to be much. I said, "Quite frankly, their whole system doesn't seem to hold together very well," and I thought the big competitor to worry about was GE. And that-- he was quite impressed with that and he let-- that was, sort of, consistent with his reading. And I said, "From a business perspective they're the guys to worry about."

But he was talking about what they were going to do and he asked me a whole lot of random questions. He was a very nice essentially-- oh, it was Fales-- Fales was the man that my dad sent my resume to. It turned out Herbert Fales was one of Tom Watson's boyhood heroes. Fales was an adventurous pilot. He'd done all kinds of stuff and he had for years just been buying IBM stock because of my dad and so he was a big stock holder of IBM and they all knew him well. And so when he sent this thing in, Tom Watson paid attention and so he had a real in with Tom. I didn't know that but in any event so it was an extraordinary bit of luck but-- so Tom had one of his assistants take me to see Jerry Haddad, the President of their Advanced Systems Development Division.

The Advanced Systems Development Division was a new division they'd just set up and so he introduced me to the president of that division and I went up there and talked with them. And at the end of the day the personnel people got a hold of me and said, "What would you like as an offer?" And I said, "Good

question but I'm not sure I ought to tell you." I said, "Why don't you tell me what you think and I think that's the best way for me to understand what you think I'm worth." And so I got a very good offer and I went there posthaste and so that's where I was -- I learned a lot but I'll tell you. I had published a book and I had taught the courses and I done an amazing amount of stuff for a young guy. This was '59. I was 32 at that point and I'd just been having fun but it was a great time, so that's where I got.

Booch: So that's how you came into IBM and so you relocated then to where?

Humphrey: We moved-- we were in Cochituate, Massachusetts before-- it was outside near Wayland, outside-- that's where we'd been living for a number of years. We-- at this point we had four kids. Our fourth had just been born. He was quite young. He was born in April '59 and so we looked around and bought a house in Chappaqua.

Booch: I got to ask if you don't mind me asking a personal question. How much did a house there cost in those times?

Humphrey: Well, we-- the house-- the first house we bought was the one in Cochituate. It was a Martin Cerel house. It was a whole new development going in. It was a marvelously built foundation, plaster walls. I mean, you don't get that much these days. It was a beautiful house. We put $100.00 down and got it-- I think the price was around-- it was, I don't know, $20,000.00, something like that. And we got it out with a 4 ¼% GI mortgage and everything. We put it on the market and we cleaned it up every way, which way, so we had-- didn't have a speck of dirt on it.

Everything was-- with four kids, it was, kind of, surprising. We patched the wallpaper anywhere where things were marked, so the house was in pristine shape. I'd redone the basement and planted a mess of trees, built a terrace and screened porch and it was a lovely looking house. I must have planted 50 trees in that place and so we put it on the market and that day we sold it for our asking price. We had somebody across the street-- it was a real estate agent across the street debating whether to ask us-- offer us more than we were asking.

So it went like a shot. We made a little money on it, not a lot and in Chappaqua we bought a house again. The house was, kind of, rundown so it-- but it was a lovely, basic house on a very big hunk of lot-- hunk of land. Yeah, it was right across the street from where Clinton-- Bill Clinton's living. It was in a reasonably nice area.

Booch: I guess so.

Humphrey: Yeah, and we bought that house-- I think we paid $42,000.00 for that and did a lot of work on that. We later sold it for a bit of a profit. We kept outgrowing our houses. We needed a house that had six bedrooms, so we got that built and we moved over to that in 63 when our sixth child was born. That was too much to get into our old house. And so we built a new house there and we moved into that. So we lived in-- and then when we finally got into our third house in Chappaqua and-- which was bigger still, a big old castle and we did pretty well on each of these. We sold them for pretty much what we had paid for them and then we moved around again. I moved up to Endwell and they moved me up as a lab director up in Binghamton and then I came back down and we moved to Darien Connecticut and before I retired we moved back to another house in Chappaqua.

Booch: Well, didn't they say that IBM stands I've Been Moved? So--

Humphrey: Most of the moves we did just because we grew. You're figuring I had-- basically two moves were caused by IBM but the rest of them weren't. So--

Booch: So tell me about the things you did and what were you assigned to do because I know you got the title later on of Director of Programming but was that your initial title or is that something that you came into?

The Brokerage Studies

Humphrey: I came into that later. There's a couple of interesting stories that led to that. The first job I had, they put me in charge of the brokerage industry studies in the Advanced System Development Division and-- so that's where I started and I had a-- I basically had to build a group. We didn't have anybody. They gave me a group that had just-- was doing some work on the-- for the stock exchange and so I remember first day I got the job I got a hold of the branch office manager at Number 1 Broadway who ran the brokerage office, went down to see him. His name was Buck Rogers, a wonderful guy. He later became a VP of IBM. A marvelous fellow.

Booch: How can you go wrong with a name like Buck Rogers?

Humphrey: I think that's true but in any event I went and talked to Buck. Buck and-- he took me over to see some people at Merrill Lynch. He said, "That's where we want to work." I went and chatted with them and I went back up and talked to the management team and we got a meeting together because I wanted to start to work with Merrill Lynch. So we got together a meeting with the division president and all the other top people and I presented what I wanted to do and so everybody said, "That sounds great." So I left the meeting, sounded like I had an okay. I called the branch office. I said, "Let's go down and talk to them."

So I went over and talked to Merrill Lynch and that afternoon we had an agreement that we were going to put together and do a joint study. So they basically agreed on it and we agreed generally how we were going to do it. We just had to put together the documents. So I got back and told the management team. They were horrified. They said, "You can't do that." I said, "Why not? If everybody agrees we can do it." They said, "But you got to go through and get an approval from the lawyers and everybody." I said, "Let's do that, okay? That's fine." But, enough. But-- so in any event we just went ahead and did it and that's one of the things that I learned way back at Sylvania before that and that is that the old Jesuit theory. It's better to ask forgiveness than approval.

Booch: I'm curious. Did you grow up catholic? We hadn't talked about that side of it. You mentioned--

Humphrey: Oh, no. I grew up agnostic. My dad was an agnostic. I did, in fact, become a Catholic a couple of years ago after all these years after seven kids brought up in the Catholic church and my wife a lifelong Catholic.

Booch: Very good.

Humphrey: So I did-- my step mother was also Catholic but we didn't go to church at all and it's just something I ultimately decided to do. We-- there's probably not time to go into that but I am now a Catholic. And I am a lector and a Eucharistic minister and all that sort of thing, so it's great.

Booch: Marvelous, marvelous.

Humphrey: But in any event-- yeah, so I started off in the brokerage industry and we actually put together the first automated system for the New York Stock Exchange ticker system and their floor trading system. We put together a proposal for Merrill Lynch on how they could make an electronic trading network. We also worked with Bache and Thomson McKinnon and we had a bunch of others that we were talking to. We were designing a system to work with them and to help run electronic trading, basically automate the way they did it, pretty much what they've ended up doing.

I remember at one point during this point they had a meeting with the board of governors of the New York Stock Exchange in Poughkeepsie and they asked me to come give them a talk. And at this time what was interesting the biggest market trading day they had as I recall was in 1928, like, 20 million shares or something like that, 19 million shares that day. A typical trading day in those days was three million shares of stock and so when I gave my talk to the New York Stock Exchange board of governors, I talked about trends and that sort of thing and I basically started by saying, "How would you handle a 100 million share day?" That kind of blew their minds. They'd never thought about 100 million share day but I presented enough evidence to show them that it was going to happen.

In fact, it happened before a whole lot longer. We're running about 2 to 3 billion share days now. So it's way up but in any event-- so I gave that talk. They later told me the stock exchange people changed their whole perception and all of a sudden they began to think you had to go electronic.

Booch: Because back then the trade reconciliation was still largely done by humans, was it not?

Humphrey: It was all by hand.

Booch: This reminds me then of the story of what became the Depository Trust Corporation. Were they in the picture here around this time, too?

Humphrey: I didn't have any involvement with them at all. Unless that was where the-- no, they-- all of the brokerage firms, they had their own secure lock up systems for all the certificates and stuff because they still had certificates and everything was done with paper and all that sort of thing. And so basically all the brokerage houses had that-- essentially a lock up secure area where they kept all that stuff but there's-- so that's what we did there. And when I was running the brokerage industry stuff, IBM was at this time designing the 360 system and that sort of thing and if I remember what the-- order it was, yeah. They-- we got a bid-- well, let me step back. We were living in the house at this point in-- I guess we just moved to a house on-- #10 Barron Circle.

The FAA Bid

Humphrey: It's the six bedroom house that we bought and it was about 1960 -- no, '62, '63, yeah, about '63. I got a call one day. This is the day before Thanksgiving in November of '63 that there was a meeting

with [IBM's] Vin Learson Friday morning in his office at Yorktown. IBM was moving its headquarters over to Armonk and they had temporary headquarters in Yorktown and he wanted me there at 9:00 A.M. on Friday. Well, this is a day off, of course, the day after Thanksgiving. So I -- but I didn't argue about that and I said, "Okay." His secretary wouldn't tell me anything about the meeting, so I just arrived in Yorktown, got to his office and there were two other gentlemen there.

There was the president of the marketing division, Frank Cary, and the president of the development division-- Systems Development Division was George Kennard and me and Learson. And so they explained that they had gotten the bid request from the FAA for an air traffic control system and they wanted me to take charge of the proposal effort. The development team had put together their approach for how to do the job, and the development division president, a VP under him, had put this together. A fellow named Bob Evans. And the marketing division guy, the Federal Region, fellow named Ralph Pfeiffer, his team had put together a proposal from the marketing people as to what they ought to do and they were at loggerheads. They could not agree. At this point my whole brokerage group and half of the ASDD crowd had all been moved over to the marketing division.

So I was in a very strange position. I was a development engineer with all this computer background and stuff in the marketing division. And so they had scouted around to find who they could put in charge of fixing this problem and getting these things resolved and they'd landed on me. So I was pulled from way down in obscurity somewhere and I'm sure my contact with Tom Watson really didn't hurt at all. So contacts don't hurt in this field and you take advantage of them when you can and I was very lucky. So in any event they went through this and they asked me if I would run the job and I agreed and I had to figure out what we were going to go. And it was-- the day after Thanksgiving and the proposal-- the technical proposal was due before the end of the year, I think before Christmas and the financial proposal was due before New Year's Day, so we didn't have a whole lot of time. And they had these two teams and they said, "What do you want?" And I said, "I want to meet with the marketing team tomorrow in Washington."

So Frank Cary said, "Okay, we'll set that up." And then I said, "I want to meet with the development team Sunday in Poughkeepsie." George Kennard said, "Okay, we'll set that up." So they did, and I went down to meet with the marketing guys and find out what they really had to have but I wanted to go there first. And so I really got to know what the heck they were talking about and why and they'd come up with a pretty impressive story. It was a multiprocessing system just like the ones we designed at Sylvania. They were having a big battle with the development folks because they said you couldn't build such a system and you couldn't program it if you did and so I knew they were part way wrong anyway.

So I went up to Poughkeepsie Sunday and discovered that they had a real distributed crowd of people all over the place. I was introduced as the boss in both places immediately, so-- and one the things that Learson, by the way, told the two division presidents, he said, "For purposes of this proposal you report to Watts." So I was reporting to Learson and I had the two division presidents working for me. So I was given *carte blanche* and I thought it was wild and I might as well use it. So I told-- I talked to the engineering manager who was putting the proposal together and engineering it and they were trying to build it out of what was going to be the Mod 50 360 machine the following April and the Mod 50 turned out to be far ahead of the other 360 machines. It had been pretty well completely developed and they'd done a lot of work through-- I'm getting some breaths from you. Is-- can you pull your mike down a little bit?

Booch: Sure. I will do so. There we go. I had some people come into the room here. Okay, I'm good. Is that better?

Humphrey: Yeah, that's fine.

Booch: Great.

Humphrey: So in any event they used the Mod 50 as the base for building the FAA machine but they had not put together the multiprocessing the FAA wanted. The fail soft multiprocessing had to be a polymorphic system. There must have been half a dozen people bidding for this thing and I'd gotten the whole story of competition. It was the number one proposal, the biggest bid IBM had ever submitted at this point.

Booch: What dollar amount were we talking about back then?

Humphrey: $100 million.

Booch: Wow, that's a lot of money even today.

Humphrey: It was then. So I-- but I told the guy-- the engineering guy, I said, "We've got to get this team together. We can't do this unless we get the marketing guys here, too." So they all agreed and I said, "Where we going to do it?" And they didn't have any room at all and the engineering guy said, "Well, I belong to a volunteer fire department out in Red Oaks Mill and they have a big dance hall there. Maybe we could rent that?" I said, "Let's do it." So he called and rented the dance hall. It had a piano in the corner. It was just a big room. And I got a hold of the marketing guys and I said, "We're all showing up there, like, ASAP." So we-- in about two days we got in there. We had furniture. We had tables set up. There was a blackboard we could slide into the middle. We had about 50 people. Even had the financial people come in and work with us.

Booch: With the piano, did you start off by singing the IBM company song which I think existed back then?

Humphrey: No, every so often somebody would go banging something on the piano … it was marvelous fun… I'd pull the blackboard in the middle of the room and bang on it. I said, "Meeting." I told them what the meeting was. Anybody who wanted would come and so our first issue was we had to agree on a strategy and exactly how we were going to design the system. So I went through that. It took a little while. We hammered it out and decided because I'd gone through this stuff, could you build the hardware or not. I said, "First of all, this contract is not for programming. We are building the machine and here's the kind of machine they want and if we don't build that machine, we're not going to win."

And so the guys finally bought that, even though the VP of engineering over the engineering guys, Bob Evans, absolutely disagreed with the design approach. He said, "You can never get it to work." It won't-- you couldn't program it, and that sort of thing. So in any event I overrode that and so we went ahead and put together the proposal. As part of the proposal the Lincoln Lab had put together a set of four programs that you were supposed to take-- whatever your final proposal was, you had to write those four programs and come up with eight numbers. The numbers were the size of the program in bytes that it would take to store and the time it would take to execute for each of the four programs. That was eight numbers and the marketing guys had put together, with the 360 instruction set, the answers and they'd gotten their best system's engineers to do it and they'd put the whole thing to together and they had eight numbers.

So I took their-- the programs and the numbers and the specs over to see Gene Amdahl who was then the head of architecture for [System/]360 and I explained the problem to Gene. I said, “Gene, this is a-- these are $100 million numbers,” because it was, like, a Thursday or a Friday. I said, “Could you just have your folks take a look at this and see if these guys have done a good enough job?” And he said, “Okay.” I said-- he said, “When do you need it?” I said, “I need it, like, yesterday.” He said, “Okay.” So that weekend he and his two top architects re-did the programs and took 40% off the time and size. Boy, that was a $100 million weekend.

Booch: So if I may ask do you remember the nature of those four programs you were being benchmarked against? What were they exactly?

Humphrey: I do not know but they were something that Lincoln Labs had worked out. This will tell you how good a system this can be to do air traffic control and it’s-- I’m sure it was basically air traffic calculations. The intercepts and all the kind of stuff you’re after to run an air traffic control network of the type we were building. So in any event, we did get the proposal in. It was a hassle and we had to-- it was printed on, I think, Christmas Eve and we had to go down to the printers and boy, we had everybody volunteering to go down to the printers.

They wouldn’t let me go. They said, “We’ll go do it, Watts.” So they did and we got the proposal out, got it in on time but we didn’t have the pricing and then we put together the pricing. Well, it turns out they’d done some fancy work with some of the memory stuff because we had standard memories and then they had a bulk memory that IBM was also offering. So our original proposal was with bulk memory and so we priced it with the bulk memory but they-- we had some basic memory-- standard memories also, and the system could be multiprocessed. You got the four processors. You got the eight memories, a bunch of channels and everything could communicate with anything and anything could fail. And, of course, we had all of these interlocks between them that were programmable so you could actually tell status and pass control and that sort of thing, so we had all that stuff in that we had designed before and I had some marvelous people working on this. They were really very, very talented.

FAA Bid Pricing

So in any event-- and the architects were working with us as well, so we had a really very, very, very good technical crew. So we got the proposal together and submitted it and when we put the price proposal together and submitted that, the FAA took a look at it and the-- I remember having a meeting with Tom Watson and Vin Learson. Vin Learson when I’d been given the job said, “If we have to bid it at $1.00, we’re-- we’re going to win this contract.”

Booch: And if I remember your pricing came in a little bit higher than $1.00.

Humphrey: Well, I was, kind of, surprised when I arrived at the pricing meeting in Armonk with Vin Learson and Tom Watson and a bunch of folk… and it was a pricing meeting. They laid out what the costs were and all that sort of thing and so I went through that whole-- we went through all of that. The top financial guy, Hillary Faw, a marvelous guy, he was there. He described all that stuff and we went through and put together the price. We proposed what it was going to be and they were-- the marketing guys were both-- they were driving for low-- a low number. They wanted to come in under $70 million because they said, “That’s where the best competitor’s going to be. We have got to do that.” And that wasn’t profitable. And so it got around to Vin Learson who was a senior VP over marketing and development at this point.

Vin was a big guy. Remember, he was my boss. And so Tom turned to Vin and said, “Vin, where do you think we ought to price it?” He said, “$100 million.” I thought, “Oops.” Way more than the marketing guys were thinking. He said, “That’s what it’s worth.” He said, “This is a marvelous system. It’ll do the job for them, that’s what it’s worth.” So actually they priced on a-- a little bit over $100 million and that’s what they submitted.

And-- so the FAA went through that and they came back and because of the pricing, the bulk memories turned out to be most of the memory in the system but the way that we priced everything which had been, kind of, you know, done quietly by the financial community and they didn’t tell people what they were going to do. The way they priced it, it turned out that the memories in the computers themselves were priced much lower and that if the FAA ended up replacing the bulk memories with more basic memories it’d actually save money, several million dollars. And so they did. So the bid came in right under $100 million and the FAA accepted our proposal. Even though we were the highest-priced bid they took it. Well, the team we had together, this 50 person team, they were amazing. While they were all at loggerheads before this, they were all together and we kept them all there.

They were all involved. We knew what we were doing and the whole idea of building a coherent team, everyone knew what we were doing. Tom Watson never arrived there but Vin showed up for several meetings, he’d show up. He’d come in and I’d bang on the screen and introduce him and so Vin would talk to the group and so we had a whole-- we’d have a daily meeting with everybody and somebody-- something would happen and there would be this excitement. They’d bang something on the piano and then they’d go announce it.

Booch: Oh, you were still in the dancehall during this time. You rented it for a long time.

Humphrey: We were at the dancehall through New Years and we got-- it was-- they said that dancehall really paid for itself.

Booch: I’d say so. That’s certainly not a piece of IBM history I knew about.

Humphrey: No. But it was-- I mean, we had an integrated group. It was just so enthusiastic and they were-- really the excitement when we got that thing in and it was amazing. But I remember a meeting right after the win. There was another meeting called with Tom Watson with me and Learson and Ralph Pfeiffer and all the top marketing people, Frank Cary was there and the financial people and they wanted to know how the FAA had been able to snooker us by getting the price down. And so Tom was really mad and he, kind of, looked around the room and he said, “Now, who’s responsible for that?” Dead silence. But finally I said, “Oh, Tom, I probably am. I was proposal manager.” He said, “Well, what happened?” So I explained it to him about the pricing and how it had happened and I said it was, in fact, a surprise that that’s what happened. And he said, “Okay,” and so that was the end of it but I was surprised. In that whole room not a soul would step up and say, “Well, no, I was responsible.” But I figured I’d better. And in a way I was because I-- Hillary Faw later told me, he said, “You could have known all of that.” I said, “Quite frankly, you were pretty busy.” So I didn’t really understand it but that’s what happened. So we won that and shortly later they had a re-organization and Bob Evans, the VP, was pulled from his job. He was the guy that I’d had these battles with but he’d been involved in this whole thing. He knew what was going on and he was put in charge of the Federal Systems Division which was also reorganized at that point. So Bob became division president of FSD.

Booch: And this would have been what year? This would have been ’65 I think?

Humphrey: I think it was ’60-- no, it was ’64. Well, the [System] 360 was announced in April. I think shortly after they had the reorganization because they’d rearranged the whole division structure. They put a guy in charge of-- one guy in charge of all development and then they had marketing and manufacturing pulled out of the separate units. It was all rearranged and the marketing and stuff-- Learson still had development and manufacturing. Marketing was under Dick Watson, Jr. Dick Watson, Tom Watson’s younger brother and I guess-- no, I’m sorry. Frank Cary was development group manager. He had stuff under him including a systems development division and system’s development had the whole 360 and everything. So they’d gotten 360 announced and there’s a story there, too, but let me finish this first and come to that. The Federal Systems Division was later asked to bid on the programming for the FAA system. Remember, “Bo” [Bob] Evans was the guy as division president now at FSD who said you couldn’t program this thing. So his division put in the proposal to program the FAA system and they won it. And so Bob’s team actually programmed the FAA air traffic control network, the en route air traffic control network which is used to this day.

Booch: Wow, well, as a frequent flyer I feel happy that you were involved with that.

Humphrey: Boy, it is the system. It was delivered and it’s been enhanced and upgraded and all kinds of stuff but it’s been working. That’s the-- that design has been working ever since.

Booch: Now, this was all still in assembly language, correct? Because I know FORTRAN was on the scene back then but that really wasn’t being used, was it?

Humphrey: No. No, that isn’t quite right because part of our bid-- the one thing-- the one programming thing we had to deliver, they had some utilities and stuff where we had to deliver a JOVIAL compiler.

Booch: JOVIAL, of course, wow.

Humphrey: A JOVIAL was behind that thing and so that was in it. I’m sure they moved to C and C++ and everything since but in any event that was in it.

Booch: Did IBM actually build that compiler?

Humphrey: No, it was contracted out. It was part of our bid. We had a contract to do it. I don’t have any recollection but that was-- and we got a JOVIAL. We were able to buy one essentially, modified to run on the 360.

Booch: Got it.

Humphrey: But-- so that was amazing. So Bob Evans at FSD actually did the programming and they did a marvelous job. It worked fine as we knew it would. So in any event-- so I’ve forgotten where I was going to go but that’s-- so that’s what happened there.

The IBM 360 Announcement

Booch: You were going to tell me a story about the 360 and its announcement.

Humphrey: Oh, yeah, the 360 announcement, they had a meeting. This was in end of March in ’64 and at this time, after we got the FAA bid in, I had been promoted. When was I promoted? I’d been promoted, I guess, in January. I’d been promoted to Director of Systems and Applications Engineering on the corporate staff and I was now reporting to Dick-- to a guy Dick Bullen who worked for Dick Watson. And I guess I reported directly to Dick Watson originally, that’s right, before Dick actually left IBM and went to be Ambassador in Paris. But in any event so Dick was my boss there for a while. I knew him. But I got called to this meeting and it turned out it was the meeting where Bob Evans and Fred Brooks were presenting the 360 to get approval to announce it.

Booch: Was this your first exposure to Fred or had you run across him before?

Humphrey: Oh, no, I’d run across Fred before in terms of-- he’d gone around all the divisions and talked to all the people about 360 and his vision of it. Fred had been marvelous. He kept everybody up to speed and trying to get everybody to buy it and he was a marvelous salesman. He got everybody convinced this was the way to go and he had a marvelous vision. And of course, trying to pull together all these disparate divisions at that time was extraordinarily important because they were all on different wavelengths and viewpoints but I’d say a large part of getting everybody to work together was Fred. And then getting the architecture behind it and getting people to realize you needed all of these standards for interfaces and needed a standard machine architecture and you had to have compatible programs and he had this vision but he was sharp enough to know that a vision without backers isn’t going to go anywhere. And he had put it together and got all the support for it and Bob Evans who was the bulldozer so that fortunately Fred Brooks was the guy who did it. I mean Bob [Evans] was bulldozing the senior management. He didn't have to fight the technical stuff. And so that was a marvelous team. They did a great job. As you may know they got a National Medal of Technology for that.

Booch: Yes Fred told me the story when I think he went for that. I guess it was in 2000 something was it not? Was it a bit later?

Humphrey: I don't remember what year they got theirs but I think it was a little before that. It was one of the early ones.

Booch: Yeah, yeah Fred told me the story that he was there at the same time Steve Jobs was and Fred said he had a problem with his Apple and Steve handed him his card and said, "Send it out to me I'll get it fixed."

Humphrey: Oh, is that right?

Booch: Small world.

Humphrey: But so I sat through the meeting where they were reviewing the proposed announcement and Fred got started with the architecture of all the technical discussion and Tom Watson got up and walked out of the room.

Booch: Wow.

Humphrey: They were all sitting there and Learson was sitting there and he said, "Go ahead keep going." They went through the whole discussion. He described it all technically what they were doing and everything else and about the time he wrapped up, Tom Watson came back in, sat down for the financial and marketing discussions. How he knew when to come back I'm not quite sure maybe somebody snuck out and told him.

Booch: So what did you think of that as a sign of him leaving during that? Were you all concerned or how did you feel?

Humphrey: I guess I wasn't too surprised. It was a very good tutorial discussion but I learned by then that executives weren't interested in tutorials. And I mean Tom had a lot on his plate. One thing I didn't-- he was a great multi-programmer. The day I met with him for my very first interview in New York at 590 [Madison Ave, IBM headquarters], the next day was the annual meeting that he was presiding over.

Booch: Oh my, busy guy.

Humphrey: Yeah. So this guy he took an hour to meet with me and I thought this guy can compartmentalize stuff pretty damn well. So that's what he was doing in the System 360 meeting. He knew what he was doing. He knew-- he was satisfied with the technology and he'd gone through all of that stuff and he wasn't going to waste time listening to us. So he just went out and did something and came back when he was needed.

Booch: Good for him.

Humphrey: And so they approved the April 4th announcement and they went forth with the announcement, changed the world.

Booch: Indeed they did.

IBM Time Sharing

Humphrey: Yeah. So I was on corporate staff as Director of Systems Engineering-- Systems and Application Engineering, which had systems engineering in the field and computer software. I was director of both of them. And the following November, November of '64, I got a call to a meeting and this is with Vin Learson and Dick Watson Jr. and they -- no I got a call from Learson I'm sorry. It was from Learson office and he wanted to meet with me on Saturday in White Plains. I told his office that I could but I had a date with my wife in New York for dinner and the symphony and I really didn't want to miss that. And he said, "Well when do you need to leave?" This was his assistant that I was talking to. And I said, "I need to be able to leave at five." He said, "Oh, okay." So the meeting was like at two or something like that Saturday afternoon. So I arrived in Learson's office at two and there was a crowd there and they were starting to present a story. And I didn't know what the meeting was about or anything. I was in the dark completely. And so Learson kept pushing he said, "Now hurry up now Watts has got to leave at five." And he had all these people there and no one knew what I was doing sitting there. And so they went through all of this stuff and Learson finally got them to finish by four. And then he said, "Okay." So they all trooped out and I was the only one left.

And so it was Learson and I and another guy who had become Group Executive, Gibson, John Gibson. He was a Senior VP of IBM and he-- and he and Learson,-- Learson had, at this point, marketing and the FSD people. He did not have development and manufacturing. Gibson had development and manufacturing and research, the research division and other stuff. And so it turned out that what they wanted me to do was to report to them and to take over a proposal again. But this time what had happened was that MIT had come up with a timesharing approach that required virtual memory. And they had been negotiating with IBM to get IBM to put virtual memory capabilities into the 360 line and IBM had flat out refused arguing that the virtual memory wasn't needed, memory can be much bigger. Amdahl was the key behind all this. He said, "We don't need virtual memory we'll just have bigger physical memory." And memories were getting bigger and so we didn't have that problem. And MIT wasn't buying it and so they had come up with their own design. And they got GE-- remember we talked about GE, they got GE to build it.

And so MIT had actually been-- had bought the GE Multics machine and system, the GE system which had the multiprocessing with virtual memory and they were working jointly with GE, MIT and GE did the design for the Multics System and Corbato at MIT was sort of the key guy to run that, the design and development of that. And they were bidding that system now, GE was, to Lincoln Laboratory and a bunch of other people. And the nervousness from the marketing people was, and that's why they were so upset about this, and the top marketing people were in the meeting with Learson, by the way, before they walked out. But we got this whole story and the Bell Laboratories was leaning toward getting the GE Multics System, as was General Motors. Essentially IBM's top 10 accounts were all tending towards going with Multics. And the marketing guys were scared to death.

This was becoming a hell of an erosion. It would get GE really in the catbird seat and IBM 360 would be in real trouble because these were the top people and Multics would be the wave of the future. So they-- there was a proposal due next to Lincoln Laboratories. IBM had already lost the proposal at MIT to GE. And so the Lincoln Labs proposal was now due and we had-- there were three weeks. I remember I-- no I guess we had a couple of months or something for the Lincoln Labs deal, it wasn't very long. I guess it had to be in like the beginning of the year or something. And they had been working on it and nobody had figured out what they were going to do.

Booch: Sure. And correct me if I'm wrong but around this time there was the phrase about IBM and the seven dwarfs because GE was one of them and there were lots of competitors nibbling at IBM's heels around that time.

Humphrey: Oh yeah, NCR was there and we had the Unisys, of course, and CDC and Philco and, oh there were a whole mess of people. So it wasn't a small band at all and there was a lot of ferment going on in the field. So in any event, I was brought into this thing and again same kind of story, "Whatever you need Watts," and you know, "go do it." The problem was the 360 was taking up all the horsepower we had. And the problem we had was we had to build the programming system. And so I got agreement that the people in the Advanced Systems Development Division up in Yorktown Heights who had actually build a time sharing system, an early one which had been worked with MIT was the precursor to Multics. And so they had this early version. And so I asked for that team. So I got that team and it was a very good folk, marvelous folk. And I got a guy as the marketing director-- Orville Wright.

Booch: What a name.

Humphrey: Yes.

Booch: From Buck Rogers to Orville Wright.

Humphrey: Orville Wright it turned out later left and was president of one of the big telephone companies and so he got a bunch of big jobs, good jobs.

Booch: Any relation to the famous Orville Wright from way back?

Humphrey: Yes, a grandson.

Booch: A grandson wow small, small world. Okay.

Humphrey: I think Orville has subsequently died. But he was very nice guy and a marvelous marketing director. And I-- at the same-- so we had this thing, we had this bonfire going, we had to win the Lincoln Labs proposal and I got to know the people up there. Very good crowd and very logical people, they were very sensible, marvelous customers. So we were putting together a bid for them and I spent quite a bit of time with them to understand what they really wanted and that sort of thing. It was pretty clear we had to have virtual memory, period. So we got together with Jerry Blaauw and a whole bunch of people and I put them in a building. I think it was called Fluellen House on the Yorktown Heights property where the Yorktown research lab is, down in the lower corner there, bottom of the hill at the entrance level. It was just an old white house. So I commandeered the main area-- well, I commandeered the house. We got this crew down there. They all commuted in, they weren't living there. I met with them first and I told them, this was on a Thursday "you're working on weekends." I mean this wasn't one of those nine to five deals at all. And so I went through the criteria that this bid had to meet and the architecture had to do the following. And so I said, "You guys figure it out and I'll be back occasionally but here's what we've got to do." And I had the top architects and designers, we had the-- the marketing people were there as well as the top architects and the engineers. And so we had a long weekend to design the system so we could make a bid. And we had the guys from ASDD group, the programming group pulled in and those were the only programmers we had on it.

Marketing the IBM Model 67

And so we did-- I come back, they put together a marvelous proposal the Blaauw Box. Jerry Blaauw came up with the design and it was a modification-- we couldn't start building new machines obviously so we had to take the 360 and bang it up. So we decided to use the model 65 as the base for the system 360 model 67. And we had the 65 engineers there with it and they went through and figured out how to do it and how to put this stuff together. So they came up with a hardware design and the other guys came up a software strategy and so we basically had it and that was after about four days. And when I came back to go through it I told them I said, "I want you to show me how we're going to be best in each of these categories compared to Multics. And so they did, they put the proposal together and the-- I mean they put the design together and it was marvelous-- it was really a great job.

Booch: Now as I remember Gene [Amdahl] was still on the scene wasn't he? He was still at IBM around this time because it wasn't until '70' that he took off.

Humphrey: I don’t remember when he left. But he was not fighting this. I mean you knew this wasn't the first time where he was going to fight this. He still strongly disagreed with virtual memory.

Booch: And I know that was one of the reasons he moved on.

Humphrey: Yeah exactly, exactly. But in any event so we put this together, we put in the proposal. It was a very simple design for the virtual memory but it was a good one. And we put it in and we won the bid. We got the Lincoln Labs bid and the marketing guys were going off and Orville Wright and his team and they were putting out fires with this system. The Model 67 turned out to take the market by storm. I mean people loved it. And it had multiprocessing we were able-- we designed the system, the 67 was the first multiprocessor of the 360 line.

So we had to have that in place so we could have multiple computers come together with a virtual memory which is a very attractive system and they had all kinds of expansion capabilities and a big deal with some of the real-time communication that you needed and everything else. So it was a great system. When we put that in, we won and we were going great guns. The programming guys did extremely well. They were up in this lab in Yorktown Heights and then all hell broke loose. The Multics people and the GE folks had decided to leapfrog the technology and so they had come up with an expanded virtual memory approach and they sold it to General Motors.

Booch: Well I'll tell you what let's leave that as a cliffhanger here and take a short break. And then we'll pick back up to find out what you all did in response to that. How's that sound?

Humphrey: That sounds fine to me.

Booch: Great so I'll stay live on this line and let me stop the recording here.

Humphrey: Okay.

Day 1 PM, June 17, 2009

Booch: Welcome back. And in the meantime for those of you who were listening to this recording Watts and I have solved all known problems during the break and we've left the unknown ones for the readers. So Watts we were talking that about of a cliffhanger that Multics was very much on the scene and quite threatening what was happening here. Let's pick it up from there.

Humphrey: Right. Well the GE people had come up with a bid, GE and MIT together to put a much more expanded virtual memory into the Multics System and it actually had a great deal of flexibility. The reason it was attractive to General Motors was, General Motors wanted to use this timesharing system for the graphic design for their automobiles. And they had a big graphic design system and the marketing people got me out to GM to see what they were doing and why it was interesting. And they were working with the University of Michigan and Bernie Galler and folks out there. And very nice folk. I got to know Bernie quite well and he was pushing this stuff real hard. And he was sort of the intellectual push behind all of this stuff for this expanded virtual memory.

Booch: I have to ask. <inaudible> graphic displays were around back then. If they were doing their CAD work on it.

Humphrey: Well they had big IBM displays. I think it was called the 2250 or something but it was a big display running off the 360 and they had them on earlier systems but they were a big part of the 360 proposal. And so it was-- they were damn good systems. They were doing amazing things with them way back then. So it was quite something that GM had a lot going on, GM Research--that's who we were working with. I didn't tell you also the Bell Labs people we worked with Ed David and a bunch of those folks. So I got to know all of those guys it was quite an interesting bunch of folks. In any event GM was really pushing us hard on this. They had to have this added memory and they argued they couldn't--literally couldn't do without it. It was pretty obvious we were not going to win the GM bid unless we could build something substantially more then what we were doing. And the marketing people were all upset because they concluded that, if GM went with GE and Mulltics, we'd lose Bell Lab and if we lost GM and Bell Labs, we were going to lose the momentum pretty completely. And so the GM win was a big deal. And so I got together with the programming guys and the architects to figure out how we would do it and that it would take a hardware change, which we were told was straightforward. But the programming guys went through it and after like a weekend's-- a long weekend's worth of study. We were strong on long weekends in those days.

Booch: It sure sounds that way.

Humphrey: But in any event they concluded that it would add about three months to the schedule. Well like a dummy I bought it. And so we put in a proposal. We won the GM bid and Lincoln Labs was very upset with the three month delay but we talked them into it and all the other customers we had a three month delay for everybody. Everybody finally bought it we got it sold in the market. They'd all do it. And so we got started on that. We had that bid. And so this was in the fall of '65, I think it was.

IBM Director of Programming

Yeah '65, somewhere in the fall of '65' and IBM was just starting to ship the Model 50 and the head of research, Gardner Tucker, got a hold of me one day and said, "Hey Watts I'd like you to write a white paper on how you think we ought to manage programming." So I said "Oh okay." And so I put one together and it was a tirade. I was really very concerned about the management of programming and how it was done. It wasn't business-like at all and we ought to have plans and go do this and that. We had plans for the TSS system, the timesharing system, but we hadn't done the real planning for the modifications which was a failure on my part because of the market pressure. The market pressure was realistic but I could have pushed them off to make a better plan and we did not do that. And that was a real failure on my part. But in any event, I wrote this tirade and gave it to Gardner Tucker and a little bit later I got this call again from Learson. And I had been promoted to Director of Programming. It turned out they had fired the previous director.

Booch: Ominous sign.

Humphrey: Well actually they fired the guy for a reason that had nothing to do with management. He was running around with his secretary. Nice guy, I knew him. He lived in Chappaqua, he had 10 kids and he was running around with his secretary. IBM had something called an open door policy and when somebody would write a letter to Tom Watson, anybody in the company could contact Tom Watson or the Chairman. I don't think they can still do it, but they certainly could then. And this young manufacturing manager had written a letter to Tom Watson saying, "My wife is the secretary working for this big executive who's running programming and she's running around with him and I can't compete with your executives when they're running around with my wife. I'd like you to fix that."

So Tom Watson had an open door investigation and discovered that the guy was right on so they fired the Director of Programming, bam. And he left and took the secretary with him, divorced his wife and married his secretary. So he went off and was doing other stuff. I won't tell you who he was. But I remember going and meeting with the wife and that sort of thing. It was sad. But so I got the job and it was I'm sure based on the white paper I wrote.

Booch: Do you remember what you said in that paper? What kinds of suggestions, predictions?

Humphrey: Well I would think I wrote about the crucial need for planning and for an orderly structured process to manage commitments and the whole nine yards. I did not have *quality* in the list at the time. We didn't have enough perception then to do that but I did really push on the planning cycle and what you ought to do. So I was called in-- I remember-- I don't remember if it was the first week or the second week. I'm a little fuzzy on the order in which these things happened. But I think the first one was a meeting with Vin Learson. Vin Learson-- I took over the job. I remember I arrived at my office in White Plains, the program director's office and I had two secretaries and an assistant and they were spending all their time sorting my mail.

Booch: Physical mail of course back then.

Humphrey: Physical mail, they would make a daily mail summary. And the daily mail summary was a brief paragraph on every document that I got from wherever I got it and it ran to about three or four pages. And I had a stack of about three feet of mail I got every day. And they told me the previous director had taken all this mail home every night and that's what he'd spend his time doing, going through the mail. I said "Well let's do the following." I said, "I don't want any mail unless it's from my boss, his boss all the way up to Tom Watson. Anybody in that group writes directly to me, I don't want copies. If anybody writes directly to me I want those letters right away and any letters they write that copy me, I want summaries of that stuff but nothing else."

And so all of sudden I started to get no mail. So I had time to do stuff and I was able to deal with the problems. I mean the previous director was paralyzed with mail. I didn't have time to screw around with that stuff and most of it was totally uninteresting. And it was a lot of people writing to me and saying, "What do I do next?" And so I'd-- I said, "If you've really got problems like that give me a ring." And so I would get those calls occasionally, "What do I do about this or how do that?" I'd been in the job, like, days. "That's your job you figure out what you ought to do next and then come tell us if we need to know or tell your boss. I'm not going to tell you what to do."

The Learson Commitment Meeting

Humphrey: And so I basically got out of that loop right away. And Learson, I think this was at the very beginning, called me in. It may have been the second week, because I think the first day I did was go out and visit several labs the first week. And I remember I met in a couple of the labs, first thing I'd ask is, "What are your commitments?" And they didn't know. Honest to goodness they knew that they had to deliver a bunch of things, they had a list of stuff they had to deliver and they had dates and they were striving to do it. But they were all basically blue letter stuff, what was announced. And the marketing announcements were what was driving everything they were all doing. No one had any plans to do anything. And I said, I said, "Suppose you were going to do the job the right way. How would you do it?" And so they'd describe a very logical approach. How they'd do it if they had plans and requirements and they'd have schedules and they'd do it all. I said, "Well, why don't you do it that way?" They said, "We don't have time." I said, "That's insane. You're saying you don't have the time to do it right, you've got more time to do it wrong?"

And so they said, "Well, yes." I said, you know, "That makes no sense." And I left it at that, I had a couple of meetings like that at great big labs. I remember one in San Jose and a couple of others. And I went back thinking about what the heck to do. Well along about that time I also got a call from Vin Learson to a meeting in my office in Poughkeepsie. I had an office in White Plains and another in Poughkeepsie. And Learson had called a meeting for 8:00am in my office in Poughkeepsie, New York with me and all my senior management. I guess it was a day or two later because they had to come in, some of them had to come in from all over the place. And so I showed up. I showed up a few minutes late. He flew up by helicopter, the bum! So I drove there and I was stuck in traffic and I got there about 10 minutes late and he was in the middle of a tirade. About 30 managers in this room sitting around the room and he was really ranting about how, "You're killing the company. You're going to destroy the business," and this sort of thing. And I walked in and was sort of sitting watching all this stuff and what had happened was the programming schedule had slipped about three times.

They were starting to deliver hardware. No one had any idea when the software would come and no one believed it. The marketing force was in an uproar. The customers were upset and no one wanted to buy

anything anymore. The market had kind of really frozen. What's interesting was we were selling the TSS Model 67 however because people believed that. So in any event-- so Learson finished his tirade and slammed his fist on the table and Learson, by the way, who was six foot four, a great big guy, very powerful, one of the most intuitive, you know, really understood stuff. He was a brilliant man. But boy he really hammered on that table. And he then said, "God damn it, I've got to have a schedule in two weeks."

And so he looked around the room and everybody looked at me. I had had the job less than a week or maybe two weeks and I was supposed to make up a schedule. And so I looked at him and I said, "Vin," because I knew him pretty well at this point, "I said I can give you a schedule today if you want it but then I'll give you another one tomorrow." I said, "If you want a schedule we're going to meet it will take 60 days." Then he started around the room one by one asking the managers, "What do you think?" And he went to every manager around the whole room and everyone of them said, "Yep, 60 days." And then he turned to me and he said, "Okay God damn it," and he slammed his fist on the table and he stormed out the door. I was the hero.

Booch: That's great. You were saying the things that no one had the guts to say.

Humphrey: No one had ever said it before. I knew Vin well enough to be able to do that because I knew it was crazy. I mean of these were—some of these labs were in France and England and there was no way in the world I was going to be able to even speak to them all in two weeks. So getting schedules was a hell of a job. So they really, you know, I was a hero. And so all of a sudden from being a bum I became a hero. And I'm convinced although he never said it. I didn't really figure it out until later.

I'm convinced Learson did it on purpose. It was a shoot out. I was the enemy-- now remember now the TSS Model 67 was the enemy to the 360. And that's what I was <inaudible>. So I was in charge of the timesharing group and that was the competitor to the 360. And so I was the enemy brought in to run 360 programming and Learson had to make me a hero. And he knew how to do it. That's the kind of guy he was. I mean when I was on corporate staff as I say I was made Director-- I didn't tell you. I was made Director of Timesharing but, you know, in this Learson meeting I had earlier and as Director of Timesharing I reported to Dick Watson it turned out. But so now-- when I was Director of Systems and Applications Engineering, Learson used to ask me to meetings in his office. And I'd go listen to the meetings and I had no idea what was going on at the meeting or why I was even there.

The meeting would end and the guys would walk out and Learson would look at me and say, "Well what do they want?" And I'd say, "Well, I don't know, here's what they said they wanted." And he said, "That isn't what they want." He said, "Here's what they want." So he could see through all this stuff, just extraordinary. I remember once he was really beating us up because we were trying to get more systems engineering in the field, system engineers, I was Director of Systems Engineering as well as Programming. And I'd go in there with a story on why we needed more and the planning. It was all in the planning cycle. And every time I did he'd have some facts and data that I don't know where the hell he got it but he knew more about it than I did. And so he was always ahead of me. So I talked to my guys about this and so they were really searching around and one of them one day they came to me and said, "Boss I got it. Here's a guy who used to work for Learson. He's in this-- in a basement office for a group in marketing and he handles all the records and all the data and stuff and he goes way back knows all this stuff. He's an old friend of Learson's." He said, "I think he's feeding Learson all this stuff."

Booch: He's your mole.

Humphrey: Yeah. I said, "Oh okay let's go see him." So we called up and went over and chatted with him. I told him our proposal and why and went through the whole thing with him and explained to him what we thought and he bought it. He thought it made a lot of sense. I said, "Okay thank you." So I went back and next time I went to Learson, I said, "By the way I chatted with so and so and here's what he said." Learson just smiled and when we finished the discussion he bought it. So I mean he was real sharp. That's what he wanted to know, could I get there, and so we were able to do that. But he was an amazing guy. So I was Director of Programming and what I figured out was the problem the guys were having with not doing it right was that they didn't have any choice. And it wasn't really their problem, it was mine. The reason they weren't doing the job right and the reason was that when all they had to do to deliver programs they only had to do two things and what would those be? Code and test right?

Booch: Right.

Just Coding and Testing

Humphrey: That's all they were doing. Everything-- and they had this enormous list of stuff they had to do. But there were only two items that were essential to ship product right now and they had one problem which was to ship product. And as a consequence if I let them ship product by just coding and testing then that's all they had time to do. There wasn't time for anything else. So I realized that my problem was I didn't have an appropriate set of requirements on what they had to do to deliver products. So I went to Frank Cary, who was then Group Executive over all the development. And I said, "Frank," I was just new in the job and I explained to him. I said, "Here's the problem we've got and if we're going to fix it we have got to plan. And so I was going to stop everything. And to make sure we get plans for everything before we re-announce. I want to open all the schedules and tell the marketing organization we will get back with schedules in 60 days." And so he bought it.

And so I did, I sent out a wire to everybody to that effect. Now the people-- the engineers could keep writing programs and testing all they wanted. But we weren't going to announce anything. We weren't going to ship anything. We weren't going to do anything, not going to start any programs. I wouldn't even fund anything unless I had a plan on my desk and they had to get back to me within 60 days and they had to review it with me, all the important plans. I couldn't look at all the details. But every important plan they had to review with me personally and I wanted to make sure it was signed off by everybody who had got to work on it. I didn't want them making plans for somebody else if they didn't agree.

And so I put that directive out and it made a hell of an impact, I'll tell you, worldwide. And so people started planning. And we had to get the manufacturing planning people-- I got agreement for them to go sit with the programming guys and show them what plans were. And they'd be consultants for them on planning. So the manufacturing community was extremely helpful in going in and showing these guys what a plan is. They didn't know. And so we got damn good plans and I reviewed them all and they'd come in and as I'd go through their review I'd say, "Look first of all I want a plan you're going to meet." Here all of a sudden I turned my hat completely around, because before we were pushing to get this thing going. But I told them I said, "Look the prior director was fired", they didn't know why but I did. I said, "The prior director was fired and I don't want to be fired." I said, "If you miss a schedule you may be embarrassed, I could get canned." I'm not going to get canned, so I want schedules you're going to meet. And so they'd go through their plans and I'd poke at them. Get them to demonstrate that they could do it, that everybody agreed with this stuff. Have you got enough resources? What's your timing? And so I

ended up-- actually one thing I didn't cut a single schedule. I lengthened some of them. And I also put a 90 day cushion on every one.

Booch: Would it be fair to say that this was the most serious planning this group had ever really done? So this was novel to them.

Meeting the Plan

Humphrey: They had never made a plan. They didn't know what plans were. We put together a course. I got Al Pietrasonta to move up from the Federal Systems Division to put it together, I think it was two weeks, I'm not sure if it was a one week or a two week course on planning. And in the next several years we put 1,000 managers through planning training. We had people from marketing and other places but we put everybody through it. It was a marvelous course. Al did a great job. He died early a number of years ago and I dedicated one of my books to him, I don't know if you knew him. But in any event-- and I talked to every class. I'd go talk on Friday afternoon. And so it was something. So these guys put these plans together and they met them. We didn’t miss a date for 2-1/2 years. Isn't that incredible?

Booch: That is amazing. How much of that 90-day cushion did you end up eating over time though?

Humphrey: Well let me come back to that. First of all before I get to there, Frank Cary, when he agreed that I could stop everything and do this and open all the schedules and everything, he said, "You're going to have to go out and give a presentation on exactly what it is we're doing for the marketing Hundred Percent Clubs. There were four Hundred Percent Clubs, those were all the top salesmen in the company. It'd be about 3 or 4,000 people in these club meetings. And so he said, "You're going to have to go give them a talk and tell them what we're doing." I said, "Okay." So I put together a one hour talk, and I walked through what we were doing. I did, and I said, "Here are the problems, here's exactly what we're doing." We had performance problems, schedule problems, all kinds of stuff. And so I laid that out, and then I laid out here's what we're doing about it. And so I went out and went to these Hundred Percent Club meetings, and told them I said, "One, we're opening all the schedules, we're going to have plans in 60 days, and here's what we're doing." So I went through, I described the performance problems and the measurement system we were putting in place for measuring performance and all the other stuff. It was a pretty damn good talk, I gave it four times, and I later had a number of marketing people, one was Mike Armstrong. You know who Mike Armstrong is?

Booch: The name is familiar to me, but I really know nothing about him, tell me.

Humphrey: He was a top IBM executive, he was a neighbor when I lived in Darien [CT] and he was a top IBM executive. He left IBM in some reorganization. He didn't get moved up when somebody else did. So he left and became president of Hughes Aircraft for awhile, and then he moved from there to become CEO of AT&T, the old AT&T. And he was the guy who ran that, and actually they ran into some real problems, but nonetheless, he was a very good guy, marvelous boss. But he sort of got trapped by the technologists. But in any event he told me when I met with him later at IBM he'd been one of the salesmen in the crowd, and he told me later he said, "For the first time we had something we could sell." So my Hundred Percent Club talk basically turned the marketing team around. They said, "Now we got something we can believe." And so they went back and they did turn the market around and it was that, they believed me, there was confidence. And although I hadn't told them what the schedules were, we opened everything. The teams did come back we got the plans within 60 days. The 90-day cushion I had

put on, the teams all said, "No, we recognize your 90 days, but we'll meet our schedule and we'll stick with it." And the first couple of releases were 90 days early. And all of a sudden the whole marketing complexion changed, I mean all of a sudden these guys were delivering ahead of schedule, and what a difference it made. And so they could sell it. It also completely changed the dynamics of the debates I had with marketing, because people now believed me. I didn't have to fight anybody if I said it was going to take another year or six months or whatever they didn't argue with me.

The Model 91 Announcement

One thing happened I didn't tell you my first day on the job I went in to my office in White Plains, and then I drove to Poughkeepsie. When I got to the Poughkeepsie office, one of the main questions I had gotten, and it was on my desk one of the things they gave me was a blue letter. IBM announcements were made on blue letters, and the blue letter announcement I got was for the Model 91 programming. They'd just announced a new program the day I got on the job. And so I called Poughkeepsie, I said "I am going to come up there this afternoon I want to meet with some folks, but I'd like to see the Model 91 programmers and the programming manager."

They said, "Okay." So I had Dick Bevier, a great guy and sort of a number two on this, and he was helping me running all the other labs and this sort of thing. So he set that up for me. I had a secretary in Poughkeepsie as well. And so I got up there and met with some folks, and then along about mid-afternoon in came a couple of guys on the Model 91. And I said, "What's this all about?" And they said you wanted to find out about the Model 91?" I said "What's the plan here? What's going on?" And they said, "Well, you know, we're just starting to work on it. I don't know, what do you want to know?" And I said "Well here's this blue letter what's this all about?" And they'd never seen it.

Booch: Oh my gosh.

Humphrey: I couldn't believe it. And so I said, "Well, who approved this thing?" They said, "Beats me." So I called the Marketing VP, who I knew, a fellow named Jack Rogers, marvelous guy, who was in charge of all of the product marketing announcements like this. And I said, "Jack, I didn't know you had a programming group." He said, "What do you mean?" I said, "Well, Model 91 programming, we're not doing that are you doing it?" He said, "We certainly are not." I said, "Why are you making this announcement?" He said, "I don't know." I said, "You better withdraw it now." He said, "What do you mean, I can't." I said, "You better 'cause I'm not doing it." I said, "It's not in our plans. I've got nothing. I don't have anybody working on it there's nothing going on." I said, "This is a totaling fictitious announcement. Kill it." So he did. He sent out a wire withdrawing the announcement.

Booch: That must be unprecedented, wow.

Humphrey: Well, I tell you, it was unprecedented, but it caused an explosion. The next day, they called a meeting with Frank Cary in White Plains, all the marketing guys, I had to be there, a bunch of people, and it was on the Model 91. And it turns out the Model 91 announcement was absolutely needed for two or three enormous [customers], I think they were NASA bids or something, whatever they were they were several big bids for the government.

Booch: Yeah I remember one of the installations for the 91 was at the Goddard Space Flight Center as I recall.

Humphrey: Yeah that's what it was, and CDC was the big competitor and we had to have a commercially announced product in order to bid. And that's the only way they could do it otherwise it would be a custom bid and they couldn't get it out there and all that kind of stuff. If it was a commercial announced product they could just bid it off the shelf, but if not they would have to use special government pricing. So that's why this announcement had been hopped up by somebody in marketing and headquarters. The engineering headquarters, the division president had actually approved it. Well, then-- oh God what was his name? He later died. He left IBM, his name was John Haanstra.

So I stepped into the middle of this explosion, and I didn't even know it. But it was a crazy announcement I wasn't going to tolerate it. My job was on the line, folks. I was nailed by this thing I wasn't going to stand for that. So this is before I'd gone through the stuff with Frank Cary on dates or anything. So in any event this meeting with Frank Cary, they were just about to start the meeting, and they were going to crucify me. I mean they had to have this thing out for this enormously important bid. In walked a fellow named Joe Brown, he was a Model 91 Engineering Manager. He said, "Before you start the meeting I'd like to tell you something." So they said, "Okay." So he explained what they called a cracked-stripe problem. It turned out over the weekend the Model 91 had developed a cracked-stripe problem, and they'd been working on over the previous couple of weeks and they'd gotten some clues on the problem. It turned out the circuits didn't work. The chips, the way they were built, developed cracks, and they had a chemistry problem, they had all kinds of stuff, and he said, "Frankly the schedule is in serious doubt we've got no idea how long it will take to fix it." I was saved.

Booch: Saved by physics.

Humphrey: It's better to be lucky than wise at times, but I was absolutely stark staring lucky. So all of a sudden this was cooled off, and thank God I'd withdrawn that announcement. If I hadn't withdrawn the announcement, we would've been committed to deliver the 91s on a date that wasn't anywhere near feasible. Turned out CDC got into terrible trouble so IBM ended winning the business anyway, but it was extraordinary. So as I say, luck is something you want to have and sometimes you're down on it. But I've concluded when you count on luck you don't get it. So in any event that was that. To move on a little bit we-- so we went ahead we did the planning and we delivered all of it on schedule and everything worked well. That was a big change. So there were two different things I had had big battles about. It was after the FAA one, I'm trying to remember exactly when it was, but this fight was also with [Bob] Evans maybe it was after this he went to run FAA. But in any event I think that could well have been what it was. So things moved on awhile, and I ran programming, and we were delivering on schedule, and we were heroes and everything else and then they had another re-organization. And they restructured everything and the development division was now given to Bob Evans. And I became Bob Evans'-- Bob Evans became my boss. This is the guy I had faced down in two different really serious crises, and all of a sudden he was my boss. I'd had battles with him over timesharing and I'd had battles with him over the FAA thing. It was the timesharing when I fought with him. I hadn't had any problems with the 360 stuff. But in any event, so they decided they were going to re-organize now and I'd been running Programming for three or four years. This is like '69, late '69.

Booch: If I may ask another name around this did Nick Donofrio come on the scene do you run across him around that time?

Humphrey: Yeah, he was somebody's assistant.

Booch: Okay.

The Development Reorganization

Humphrey: But during the course of this whole thing I'd been promoted to VP of Technical Development where I had programming and systems architecture, and all of that. The 370 line was coming along and it was all before Bob Evans came, and they re-organized. But the 370 line was now moving along, 360 was being delivered at a great rate and so they were now looking at the next step which was 370. And so Bob Evans came in as my boss, and he was now looking at how to re-organize the development division. But one of the very first things that we went forward with, Don Gavis, who was now reporting to me running the 360 programming, had replaced-- Don Gavis had replaced Fritz Trapnell, who left IBM and went to work for Amdahl as a matter of fact doing 91 programming. I later ran into Fritz. But in any event the, where was I going here? Yeah, Don Gavis came to me, and said he and his team had concluded they needed to put virtual memory in the 370, and explained it. So Bob Evans had just come in and we went to Bob and explained why we needed virtual memory in the 370. And we went through the whole thing and he bought it. He was a decisive guy, he was a brilliant guy and decisive guy. He had a problem that a lot of really brilliant people have in that he tended to view his opinions of what ought to be as fact. And so he had these facts that he was convinced of that turned out not to be supportable. And then again later, turned out we're going to take longer than I thought, but let me finish this. We may have to – the IBM story is going to run a little longer then I thought it would.

Booch: That's fine.

Humphrey: There are stories here I keep thinking of that are important that I think I ought to get written down.

Booch: Well, just to be mindful because I want to make sure you have time for a break and so we should find sort of a stopping place here and reconnect on the IBM stories? And actually that might give you some time to think of those stories you'd like to cover as well too because you left IBM in what year?

Humphrey: I left IBM in 1986.

Booch: Okay, so we've got about a decade yet to cover.

Humphrey: Let me finish one piece here, and that is: Bob Evans decided to re-organize IBM, which he did. But he wanted to re-organize it with each engineering department having its own programming group. So he's going to break up this big programming thing. Part of the problem had been that I had not allowed any programming product to be announced without my approval. And for me to approve it I had to have a plan on my desk. Well, it turned out that every product, hardware or software had software in it. No one could announce anything, even a feature without my approval, and that really frosted everybody. The power structure had completely changed. All of a sudden Humphrey was over every announcement; you couldn't make an announcement without his approval and I put some real tough constraints on it.

And of course we weren't missing dates anymore. So this had the whole community up in arms, and they had to fix that and the way they fixed it of course was to break programming up and give every engineering manager-- so they made-- the debate Evans had had, he had a bunch of us in a big meeting of lab directors, and I was there. How do you organize programming how do you organize the business? And my position was you've got to organize it around programming systems. He didn't buy that. Their conclusion was we're going to organize it around hardware systems. They had large systems, intermediate systems and small systems and they had system names and the System 38 and the system this and that. I mean, so they had all this stuff and everybody got their own programming. So that's what they did. And so all of a sudden I didn't have a job. And so Bob decided I was going to run the laboratories in Binghamton. So that was my next job, and he talked me into doing that. So that's where we left. Now the 370 had to have virtual memory and that was what we were going to do and we hadn't designed it yet, and the hardware guys had to go do that, and so we'll come back to that, and I took over the Endicott lab next. Okay.

Booch: That sounds like a great stopping place. My goodness, let me stop the recording here.

Day 2 AM, June 18, 2009

Booch: And we're back live now. So welcome to part 2 of an unknown number of parts here. This is Grady Booch with Watts Humphrey and now here we are in the morning of Thursday, June 18th [2009] continuing our most fascinating interview. As we were parting the call yesterday, your wife, who I was delighted to have on this as well, too, pointed out that I had forgotten to look way back in your past and apparently there are some fascinating people as we reach back down the chain of DNA that became Watts so tell me a bit about that.

Family History

Humphrey: Okay. Okay, well, let me first mention-- I'll talk to my Humphrey grandfather. My grandfather, who was the first Watts Humphrey, my father, the second, I'm third and I've got a son who's the fourth, he's got two daughters, by the way, so the line stops, I think, but my grandfather Watts actually was an officer in the Civil War, which is kind of hard to believe, the Michigan Cavalry. He was a lawyer in Saginaw, Michigan but he was an officer in the Michigan Cavalry and I don't know what battle but he was wounded by a chain and ball canon. He was riding his horse. The horse was taken out and his leg was rather badly damaged. He came to in an army tent and the surgeon was approaching him with a saw. What you may not know is, in that timeframe, fatalities from amputations were close to 100%.

So he asked the doctor, he said, "What are you going to do?" and the guy said, "I'm going to amputate your leg." And my grandfather said, "No, you're not." And he said, "Oh, yes, I am. I have got to amputate your leg." Well, my grandfather pulled out his revolver and said, "No, you're not." <laughter> So they did not amputate his leg. It later turned out to be his good leg. He had a problem with the other leg. Without that, we wouldn't have had the whole Humphrey clan. So that was the grandfather Humphrey story. His second wife was my grandmother. He married again when he was older but it's hard to believe that my grandfather would have been in the Civil War. So generations go back mighty far and fast.

Booch: They do, indeed.

Humphrey: My father's elder brother was George Humphrey and George Humphrey worked for the M.A. Hannah mining company and he ended up being the first secretary of the treasury under Eisenhower. That was when Eisenhower was elected in, what was it, in '52 or '56, I don't know what year it was. Truman was elected in '48, that's right, so it would have been '52 when Eisenhower came in. He was secretary of the treasury then. I remember once going and visiting them in Washington and my aunt said, "Would you like to go see your Uncle George, he's in his office?" And I said, "Sure." So she had a car pick me up and it was U.S. license plate number two. And so here I was, this little guy, college guy in the back seat, I was still in graduate school, and every time we stopped at a light or anything, everybody was looking in the windows. They didn't have tinted windows. But he pulled up between the west wing and the Treasury and there was a little door there and he said, "Why don't you just go in there?"

And just to tell you how different the times were, there was no guard or anything there at all. It was just a door there. So I went out and opened the door, it was down a couple of steps on the Treasury there on the side closest to the White House, so I went down a couple stairs, went in and went up these stairs and knocked on a door and a voice said, "Come in". I walked in and I was in my uncle's office. No guards, no

nothing. It was the view out there over the Mall and the White House and everything and we sat and chatted for awhile. A wonderful guy but the security and the whole situation was so different. But, in any event, he was a great guy. At home, he drove a Chevrolet. They didn't have fancy cars or anything. But he, as I say, was the first secretary of the Treasury. On the other side of the family, the Strong family, my mother's maiden name was Strong, and her father was Benjamin Strong. He was a financial wizard. Turned out, he'd been involved with a whole lot of stuff, starting way back in about 1900. 1909, he was involved in a crisis with the Morgans and everything, a banking crisis. He actually never went to college. His older two brothers both got degrees but the money ran out. He ended up, in 1913, when Woodrow Wilson signed the Federal Reserve Act, establishing U.S. Federal Reserve system, my grandfather was the first governor of the New York Federal Reserve Bank. That was the number one bank and so he was involved in all the crises and the war, financing for World War I, the reparations after World War I and all of that.

There's a book written about him, *Lords of Finance*. Actually, during a lot of his travels, he contracted tuberculosis and so he suffered terribly from that and he actually died when I was one year old. He was 55. As they say, if he had not died, the whole financial system, the results might have been quite different because, in the U.S. government, there was no strong man to take over. He had been a strong man, running the Federal Reserve and there was a jurisdictional dispute between the Federal Reserve governors in Washington and the board that he chaired in New York. As long as he was the chairman, he kind of overrode everything and he was running it. Everybody sort of backed down and he did it. He knew what he was doing, he was pushing through, he was aggressive. So you don't know where I got it, right? But, in any event, he died right at the crucial point in 1928, October 28th.

That's when the banks collapsed and the-- there was nobody there to do what Bernanke and these guys are doing now because the people on the board in Washington were all political appointees and knew nothing about banking. All they knew about was you had to have balanced budgets and you wanted to make sure there wasn't inflation. So they were back to traditional controls and they did essentially everything wrong. So that's one of the key things. So it was a tragedy that he wasn't there but he died of tuberculosis and complications that came with that. So there's a little bit of a family anecdote.

Booch: Amazing. Banking is in your blood in a variety of ways, isn't it?

Humphrey: I guess so.

Booch: Your ancestors from sort of the management of it and you from the automation of it.

Humphrey: Yeah. That's right. Connected in multiple ways.

Booch: It is. Well, I'm sorry, go ahead?

Humphrey: I just want to say, in returning to the story, I had gotten to where we were talking about, I believe, Bob Evans coming in to the re-organized divisions and basically my career running programming was winding to a stop and, as I thought about that, a whole slew of anecdotes I need to drop back in here with appropriate dates for things that had happened before that I probably should mention.

Booch: Absolutely. So let's rewind to around '64, as I recall, where some of those anecdotes begin.

The IBM Board Meeting on Program Pricing

Humphrey: Right. In 1964, I was on corporate staff. I was Director of Systems and Applications Engineering. At that point, the IBM 360 was just getting started. I think it was '64/'65. I'm a little vague but I think it was '64, RCA was making noise about their system, I don't know if they called it 370, whatever it was, but the RCA system, they were going to make a system compatible to the 360. RCA Specter or Spectrum-- I don't remember what it was.

Booch: The Spectra 70 is, I believe, what they called it.

Humphrey: Spectra 70.

Booch: They announced it in '64.

Humphrey: Right. Okay. That was the big concern, that IBM had, that RCA was going to steal a march on them and so I was put in charge of a task force to figure out what do we do about it. And you still going through here okay?

Booch: Oh, absolutely. Very well.

Humphrey: Okay. Okay. So we went through that. We had a technical study. Had the lawyers involved. There was a great deal of concern about antitrust from the legal community and I remember we had basically no notes, so it was all talk and all on chalkboard, didn't have many whiteboards then, so we were all talking about it like that. They finally concluded, the lawyers and everybody concluded we had to present the story to the IBM board of directors. So I've forgotten who the top lawyers were at this time but they advised me that I could make a presentation but I couldn't use any charts and the board members couldn't make notes.

Booch: And, of course, when we speak of charts back then, you're talking the old overhead foils, if I recall. Certainly predating the days of PowerPoint.

Humphrey: No, these were actually paper flip charts.

Booch: Oh, my goodness. Even before foils. Okay.

Humphrey: Right. And so I couldn't even put together-- we'd usually go in with great big wads of flip charts, like three and four inches thick and backup charts and stuff but I couldn't use anything. I had to just talk. And so I had 13 points to make and I remember going in. We were talking about what the various choices we could do fighting this and clearly one of them, which turned out to be the only one that appeared to be viable from all sides, was we would have to price our programs [i.e. charge for software]. The other issue, on the other side of it, and I went through all the choices, that was the issue. So I arrived in the boardroom to give this presentation and I told them that I had 13 points to make, that I would use no charts. I had a blackboard brought in so I could write with chalk. I advised them that they should take

no notes so I would have to go through it this way and we'd have to just inform them but we could not have anything in writing. Nothing about this meeting has ever come up since. Not in any of the depositions, nothing. It never showed up so I don't think anybody knows about this meeting other than the people who were there. In any event, I did go through the whole thing about what each of the choices were and what the consequences were. I did it all from memory and I did get all 13 points. I was sort of really kind of worried that I had missed one or something but I got them all. I remember one of the board members, at the end, really-- they applauded me. I got them all. But at least we got across the point there, with a rather complete background, as to what the consequences were and the only real choices. People really could make compatible systems. We had no real way to protect it.

Basically, it said we had to start thinking about pricing programs. As the cost of programming went up, there was no viable alternative. That was in 1964 so that was very, very early. I think it's just as well we did it because the whole community was sort of ready and this was something we ultimately had to do. The next anecdote, if you remember, I talked about early on, I'd had the crisis with the FAA and then the following year, this was in November of I believe '64, I had the timesharing crunch where the 360 was out and we had the problems with Bell Labs and all of that sort of thing. Well, at that point, we were going ahead and developing the timesharing system. We'd gone through and had the upgrades.

<telephone rings>

Booch: I can stop the recording if you wish.

Humphrey: I'm getting a fax.

Booch: Okay.

Humphrey: Okay. But I will probably call back again which would be a pain so I may have to wait for it to come on. So we were developing the time sharing software. This is after we had the GM battles and all of this sort of thing. And there were the debates about how things were going to be organized. At that time there was the division president, of the development division, I believe was John Haanstra and he's the guy who later went to GE. But he was the development division president reporting to Dick Watson. And I was reporting to Dick Watson. And everybody was pushing to somehow get the timesharing stuff put under development and pull it out of corporate staff and move it where it ought to be and get it in place. And I remember meeting with Dick Watson who was then a Senior VP of IBM. And he and Vin Learson were sort of parallel VPs. He had the development division and Learson had marketing.

And so Dick was Tom's younger brother. I was in this meeting with him talking about organization and he looked to me and we were talking about this and I made some comment, "Well, if were you here's what I'd do." And Dick Watson absolutely exploded and I couldn't figure out what it was. I used a figure of speech. Well, the "if I were you", struck him as sort of presumptuous on my part. Here's this guy who thought he could be me. And I thought, my lord this guy had a terrible image problem. And he obviously did. He drank quite a bit but he was a brilliant guy. He became U.S. Ambassador to France in Paris and rather shortly after this he moved over there. But I really got on his bad side by that meeting. And so he and I were not on good terms at all. But he actually left IBM and went over to Paris. And I remember the stories were that when he called a meeting for 8 o'clock in the morning, at 8 o'clock they'd lock the door. And if people weren't there at 8 o'clock in the room they couldn't get in. And so he really tightened up the way the embassy ran in Paris. And he got into all kinds of flaps with stewardesses and stuff. And he

subsequently died from a fall. He fell downstairs. The rumor was that he was drunk. But it was sad because he was quite a guy but he was really badly overshadowed by his older brother Tom, but a very capable guy. And really a very nice guy except for the problems of his ego, I guess. The next anecdote is when I took over programming in 1966, if you remember, we were in a crisis shape and we got everybody to put together plans. But I had no way to track progress.

Phase Plans

And so what we did-- earlier on I had studied a lot of the IBM procedures to figure out, this is when I joined the company, what they did. And they had some earlier procedures from the old systems development division or I forgot what they called it now but it was the large system division, one of the hardware divisions at the time. And they had something they called phase plans for development. And it was a very orderly procedure that they had in the procedure book. People weren't really following it much and I didn't know that. But the procedure book was rather nicely laid out with about six steps on how you actually develop a program or a project.

So I adopted that for programming and required everybody to develop their plans against certain phases. So you have an early feasibility part. Before that you get feasibility ideas and then you get real project funding. You could name a project. And then you had to get certain stuff before you could announce the program. So there were various steps you had to go through and things you had to accomplish before then. Not a lot, it wasn't too much but it did require certain things and certain approvals and certain reviews at each step. So I put that in place. And so the minute we got our schedules in I had every project list their planned and actual dates against each standard process phase and track it. They'd track where they were, which ones they had finished and then when they were projecting a date, they'd have a projected versus the plan dates and that sort of thing. We put that together and it was pages of stuff. So we had these projects, as I said, 4,000 people and I think I had 15 development labs at the time and dozens of projects in every one.

So there were hundreds of these things with all of these phase dates. I put that together and I remember early on in one of my reviews with Frank Cary who was in the division-- I'm sorry, the general manager overall all development and manufacturing. He was a Senior VP at the time. It was before he became president and chairman. But I remember going in and reviewing it with him. And he was really impressed. You could see exactly where everything was, and we could track it and we could flag things that were late and it was marvelous and it worked. And we did not have sort of wild guesses in there. We were putting in actual progress and dates. Also at that time in 1966, I was, as I say, I was Director of Programming and I went out to the IBM annual meeting, stockholder meeting and it was near San Francisco in that year.

And so we went out there and at the meeting I remember Tom Watson stopped me at some point and said, "Watts, what are we going to spend on programming this year?" And our budgets had been growing like 40 percent a year. And I think the projection was it would be 60 million, the early projection, but as I had gone through the numbers it really had looked like a lot more. And I told him, I said, "Tom, I'm afraid it's going to be close to $100 million." So he actually referred to that in his remarks. He said, "I talked to Vin Learson about what we'd spend and Vin said we'd spend about $40 million. And then I talked to the division president and he said he thought we'd spend about $60 million. And I just talked to the director of programming and he said it was 100. I hope it doesn't keep growing," or something like that. But they gave us what I needed.

Booch: Let me ask you, if I may ask a question sort of about the nature of programming back then, tell me the teams, there was no sense of outsourcing back then. So was it fair to say that all of the programming staff was in the U.S.? Or did you have international groups doing it? How large was the average team? What kind of tools did they use? Can you relate upon that a bit?

Humphrey: Sure. The big lab was in Poughkeepsie. They had nearly 1,000 programmers. They were developing OS 360. And they had the control program work. They had everything. It was basically there. That's where the architecture work had been done. That's where the offices of all of the executives were. And the problem we had was that IBM had a ground rule that you couldn't get more than so many people in a geographical location. So we had headcount ceilings in every location. And Poughkeepsie had been growing and was right smack against the ceiling. And we were pushing it, every inch. And so the location manager kept kind of cutting back the amount of programming we could do in Poughkeepsie.

The Data-Management Interface

So we had to move stuff out and so a lot of stuff had been moved out. And I remember one thing we had set up and this was ahead of my time, they had moved the whole data management stuff, file support, all of that out to San Jose where the file hardware development was done. So we had the IBM San Jose software group which was growing and that's when they were building… I can't remember what it is. Was it Santa Theresa? I've forgotten now the IBM-- they built a big software lab out in San Jose and I've gone fuzzy on the name of it. But in any event so we had built a growing group out there, I'll come back to a comment on that in a minute.

Booch: Would that have been Almaden or San Jose where that was, does that ring a bell?

Humphrey: It was in San Jose, but it was a new lab they built. It wasn't Almaden, which I believe, there was a research lab there. I keep thinking Santa Theresa but that doesn't make sense, though. I don't think that's the right name. But it was an IBM software lab. It was built specifically for the software. It was really very well done. They set up with a whole series of wings where you have projects, everybody is sort of grouped together. It was very well done and it was a fine operation. We also had a group in Time Life. We had a lab in Kingston and New York City. One in Raleigh, North Carolina. We had one in Rochester, Minnesota, Boulder, Colorado, Tucson, Arizona. We had a small lab, more researchy in advanced development up in Boston at MIT. And Tom Watson had been a bug on international. So he had required us to set up labs in Europe. And they had done all of this.

This had already been done before I took over in '66. And so we had laboratories in England, Hursley, England, that's where they developed the PL1 compiler. We had one in La Gaude, France. I don't remember what they had there. It was some communications stuff or something. They were smaller. The European labs were like several hundred people. The U.S. ones, the big ones were several hundred up to a thousand. But we had it in Hursley, England, and actually it was La Gaude in France. We had a very small research sort of lab in Vienna. We had a fairly big lab in Boeblingen near Stuttgart. And a modest sized lab up in Stockholm. So we had labs all over the place. And this is before international-- you couldn't even-- even telephone communication was shaky. And so I spent a lot of time on the road going meeting with these various people in terms of seeing what was going on and that sort of thing. So I had multiple trips to Europe and with seven kids, my wife would go with me and they would spend the week going around to various cities. We'd take two of the kids at a time. So I'd meet them on weekends in places like London and Paris and Vienna and Rome. So we got some advantages of it but it was-- we had

stuff everywhere. It was really horrendous. And it was a terribly difficult collection to manage. We had to keep sorting things out.

Booch: Was there anything in Asia around this time, yet?

Humphrey: No. It was all in Europe. And the issue that I had when I took over and I began to look at the plans, we had FORTRAN compilers everywhere. Everybody was developing all kinds of random stuff. So when we started getting plans we sort of-- if you wanted a sort developed you find somebody that had some people available. And so we stamped that out. I said “No, we’re going to have missions.” And so we put the sort lab-- we originally wanted to put it over in Stockholm. It ended up, actually, with some storage experts in Poughkeepsie originally and we had to bring it back. That brings up another yarn I’ve got to talk about on lawsuits. Let me put that down here. That would be about the lawsuits. But in any event, we had all of these different labs all over the place. And typically, as I said, the big lab was Poughkeepsie with about a thousand [people].

Kingston was about 500 or 600. Boulder might have been close to that. The others were a bit smaller. And San Jose was also a big one, 500 or 600, something like that. And so that’s where we were. And one of the big issues we had was how do you pull this stuff out because when they had moved the data management work out of Poughkeepsie out to San Jose, the system was not designed to have a totally separate data management system. And so there were all kinds of interconnections. How do you define the interface between what’s the data management and what’s the control program? And no one could agree on that. So they came to me with a big battle. And this was like in the first few weeks I had taken the job and they had these warring camps. They didn’t have an answer. I sent them back and I told them-- they had a big meeting with me in my White Plains office with a whole bunch of guys from each location and they couldn't agree. So I sent them back. I said, “Come back with a proposal.” I said, “Put together a story. I can’t make this up.” So they did. And they went back and they came back with a meeting a few weeks later and they had two proposals, the proposal from San Jose and the proposal from Poughkeepsie and they presented them both to me. I was really annoyed because they had to make a decision, we couldn’t just sit there. Neither proposal satisfied the other one.

And I said, “Here’s what we’re going to do.” So I made up the answer. I knew least about this of anybody in the room. And I said, “Look, here’s what I think you need to do.” And they had gone through all of this stuff and so I carved up the baby. And I said, “You either go work that way and make it work or agree on a better way and tell me.” I said, “We can’t sit here forever so that’s what you’re doing to do.” So I designed the interface. I’m sure they changed it quite a lot and it’s probably way different now but I couldn't believe it. I knew the least about it in the room and I’m the guy who actually came up with the answers. But there we are.

Booch: You mentioned FORTAN popping about around this time. Did you have any interactions with John Backus as well?

Humphrey: I did not, actually. The only way I got involved with John, I ended up buying a house he owned. But I never had any connection with John, which was my loss.

Booch: And this was in your march to get bigger and bigger houses as your family was growing, it sounds like.

Humphrey: That's right. That was our third house in Chappaqua, a big castle that John Backus had owned. It was a gorgeous house.

Booch: Well, that's a connection unto itself. It's good to know that you owned his house, cool.

Humphrey: I did get to know Jean Sammett who was a researcher at IBM, a language researcher. She actually worked with me at Sylvania earlier on so I knew her from there so I've known her all of this time. But in any event, so that was the story with the programming, the lab, as I say my roll, and why we included planning.

The Fortune Interview

Let me move on to another situation. This was the year I took over programming. They had these programming sort of technical meetings, symposia kind of things where they get a lot of the technical people together for a meeting and they would share presentations and that sort of thing. And so I was going to go up and be the host of this thing. I had just been in the job for a few months. But I was to go up and meet with them there and Tom Watson was going to come and give a talk. So I was supposed to meet him at the airport, but I couldn't meet him at the airport because I had-- they had set up an interview with *Fortune* magazine that wanted to interview me and that was set up back in White Plains. So when Tom arrived I was in White Plains and somebody else met him in my car and it turned out I had a spare tire in the trunk and they had trouble getting Watson's baggage into it But the guy apologized for me not being there because I was doing the *Fortune* interview. And Tom Watson absolutely blew up. Because there had been some quotes in *Fortune* that he had objected to and he basically said, "We're not talking to *Fortune*." And of course, I didn't know that. So I had the interview and the people called me and said Tom had really blown up. So I called Dean McKay, who was the VP of Communications who had actually set up the interview, but fortunately he calmed Tom down but that was an explosion at the time. So actually it was a big article on the 360 and I was interviewed as part of that in *Fortune* magazine. And what was a little bit odd was that I have a bigger picture in there than just about anybody else. Also, as part of that same symposium, we had been working on improving the configuration management for the whole system and there wasn't a good configuration management system. So we had chartered Bob Rutheroff who ran the lab in Boulder, Colorado, a software lab. And his job was to put together a configuration-management control system that we would use across the whole software community. And so he put this thing together with the specs and everything. It was a very impressive story. So I remembered in my talk to the community I said that Bob Rutheroff was going to give a demonstration out at the pool this afternoon of his new system. He was going to walk across the pool because that's what it sounded like, it would do anything. So that brought the house down.

So in any event that happened. Also I think towards the end of that year we had a meeting-- well, the TSS system had gotten into trouble, as I mentioned. We had gone and added a whole lot of function to it. And instead of coming out three months late it was about six to seven months late with performance problems and everything. And so basically the system 360 was coming along and people were starting to buy it and they were happier with it. And so people had switched back from TSS to 360. So 360 was now starting to go full tilt. And the management decision, the division presidents and that whole crowd all said, "No, we're going to kill TSS, period." And I objected because I thought it was a system we should have available but no they were going to shut it down. So they did shut it down. It cost us about $30 million. We did actually get the system running and it was installed in a few places with the Model 67 but it was stopped. But it was the early virtual memory system and it was really a very good responsive system but it was not

compatible with 360 and that was a real problem that people were concerned about and it was out. So in any event, I remember meeting in the board room I was talking about software and software phase plans and the whole thing and Tom Watson interrupted me at one point, I had gone through what the phase plans were and when you announce things and when you do various stuff and he said, “Watts, I’m confused now, you did a marvelous job with the FAA and you’re probably the best guy we’ve got to run software.

Okay, so Tom had interrupted me and said, “I don’t understand. The FAA was such a tremendous success and you’re doing so very well with the 360.” He said, “How come TSS was such a disaster?” We had just closed it out at a $30 million loss. So I explained to him, I said look here’s where we announced the 360 schedule. And I showed him we had running code, we had a whole lot of stuff in place at least the beginning code and we had plans and the design was done, et cetera. And I said, “And here’s where we set the schedule for the TSS,” and it was way back at the beginning before we knew anything. And I said a big part of the problem on controlling this stuff is announcing things you don’t know how to build. We didn’t have a good foundation for a plan. So Tom understood that fortunately. He was quite a guy. He could be really tough. I remember an executive making a presentation to him and he actually reduced the guy to tears at one point. But he was logical. And if you could understand what he was concerned about and really get to the point he’d switch and he was great. So that was that. And I put down a note here on John Haanstra and Chuck Branscomb. We had the whole division structure-- when I was in programming and moved over at the very beginning when I got in there I was-- I think I mentioned to you the Model 91 programming problems.

Booch: Yes.

Humphrey: And that was like at the very beginning. And John Haanstra was the president of the division at the time. And he had been president of the small system division the old, I think it was GPD, General Products Division, and there was a DSD, Data Systems Division something like that. And George Kennard was the president of data systems division and that’s when I was in the FAA thing. And George was a prince. I worked with George multiple times. We still exchange Christmas cards, wonderful guy. And John Haanstra I didn’t know as well. He had been in the penalty box when GPD was canned, was broken up and put together in the center. But then John ended up running the big overall system division. He took that over. And he was a wild man. He had this enormous organization and he thought he could do anything. So he would basically make up announcements. He’s the guy who basically said, “Announce the Model 91, we’re going to get that business.”

Booch: Watts, I’m sorry to interrupt here for a second. There’s someone pounding on my door and I need to get it. Let me stop the recording for one second. I’m sorry. Hang on…. Now we’re back on.

Humphrey: Okay, well John Haanstra he was quite a guy. I mean he would call meetings at two in the morning. The place was a zoo. And he basically, as I say, he was very aggressive. He was moving fast, pushing hard but he didn’t seem to understand much about programming. He was a very good hardware guy. So he actually was pulled out of the job and left IBM and actually ended up taking over as head of I think it was GE’s computer operation and went down to Phoenix, I think it was that area somewhere. He also was a private pilot and he had a twin-engine plane he used to fly around. He wasn’t there a year when he was flying with, I believe, some of his family and the plane crashed. No one really understood what happened but in any event… so he died young and he didn’t continue. But he was a marvelous and capable man but he had no concept for any limitations on programming. He didn’t really appreciate it. And

he was a big part of our getting tremendously overcommitted. At another point while I was running programming development, you remember I talked about pricing earlier.

Booch: Yes.

Program Pricing

Humphrey: I already talked about the board meeting on program pricing. Now, a corporate committee was set up to actually study how we ought to price programming [software]. And people were now getting very concerned about the Department of Justice, the question about whether they were going to file a lawsuit or not. There had been a lot of discussions by then. They had been poking at us. And the lawyers were very concerned and the issue was *bundling*, basically, connecting one product to another. So if you want this product you've got to buy that one. And this is essentially what Microsoft is doing now. And the lawyers at that time told us that if you are judged to be a monopoly, that is what they say is *per se* illegal directly counter to the antitrust laws. And I often wondered how Microsoft kept getting away with it. Of course, they are getting away with it here but not with Neelie Kroes, the European Union's competition commissioner. So that is a problem and the Europeans are nailing them. But it is a problem, an antitrust problem where you tie one product to another, and so if you want this product, you've got to get that one.

Booch: Well, back then primarily the lawyers, the legal system didn't really understand software very much. What was sort of the attitude that you saw among that community? Did they really understand software? What was their perspective?

Humphrey: They didn't and I'll come back to that.

Booch: Sure.

Humphrey: Because I ended up on corporate staff later. But at this point, this is a little earlier. We hadn't been sued yet. And so I'll come back to that but actually there's a meeting I need to talk about but I'll come to that in just a minute. And that's not on my list, that's a good point. And we put that in here after number seven, which is meeting with lawyers.

Booch: Can't forget the lawyers.

Humphrey: This was a meeting on programming pricing in about 1967. Okay. But Howard Figuroa was from corporate staff, a finance guy, basically, very sharp. He was brought down. I was on the task force. And there was a marketing guy and a hardware executive. So there were about five of us, I think, in this meeting. And we were a task force. I think we spent a week or maybe two going through how would you price programming? And the question really was fundamentally how do you protect the priced asset? And one big debate was whether patenting was feasible, practical, could we do it. We concluded it was not practical. Would the patent office be able to handle it. It was too confusing. I still think that the patenting of a lot of the software is nuts. It long-term shouldn't hold up. It's extremely hard to do anything with that. And so we, basically, concluded that the fundamental choices were between trade secrets, patenting and copyright licensing. And the patenting, as I said, we felt was extremely hard to isolate it and to deal with things properly and all the featuring and all of that stuff. It would be a mechanical nightmare. And I think, for an operating system, no one has tried to patent the operating system, and I think that's valid.

Otherwise, you're patenting little pieces and ideas which is extremely difficult. The trade secret idea was very attractive in principle, except the problem was once you lose it, it's gone. And so a secret is great as long as it's a secret. And the issue was the practicality of a widely-used product, how are you going to keep it secret and then, all of the details? And we concluded that was not practical. That trade secret protection might be okay for little stuff, but not for big systems where people had to get into them and do stuff to them. And would, whether you wanted them to or not. And so we finally concluded the only real thing you could do was to have a license so that people could use it. But that meant you were licensing it to them. You weren't selling it to them. And so we had a real problem there, because it was essentially our product we were licensing people to use.

Booch: Because back then, the hardware itself, what was the legal mechanism whereby IBM got revenues? You were renting it to companies, was that the mechanism?

Humphrey: Well, yeah, that gets back into a bit of history. Let me come back to that in a minute.

Booch: Sure.

Humphrey: Let’s finish this. And that's something that I ought to talk about, because it was a big deal. I got involved in that and more in my corporate staff job in '72. Yeah, let me put that in down a little bit later, which is the IBM Business System. And let me call that in 1972. Yeah, that'll be a little bit later. So let me come back to that one. There's too many questions here.

Booch: Well, there's a lot of fascinating history. I mean, you guys were breaking ground.

Humphrey: I was involved in a lot of it. I probably ought to even mention the Iran situation. I was involved in a lot of that stuff, too, Iran and the International stuff. Where would that be? That would also be down here in about '78. Yeah, let me put in that 1978. This is Iran and China.

Booch: Oh, that would be fascinating to hear, yes.

The IBM Law Suits

Humphrey: Yeah, I could go on for days with this stuff. There was so much that was such an enormously rich experience to go through this stuff. Okay. But in any event, so yeah, now, I was now at-- we had the boardroom discussion. And that was in '63. Let me skip back a little bit. A little bit earlier than that I think I had been running programmers, [since] probably, about 1967. The lawyers would tend to get very interested in programming. And Tom Barr and the people at Cravath, Swaine and Moore, and the IBM lawyers, they all wanted to know what programming was. And about 1967, they arranged to have, like, a two- or three-hour meeting with me on what is programming all about. My staff put together this presentation, again, flip charts. I must've had 80 flip charts. And just walking through what programs are, how they're developed, how they work, the whole nine yards. And so I started going through this. And lots of questions, and it was pretty obvious, real quick. I mean, I had a room full of people. Tom Barr, I remember, was sitting back in the corner. I had met him. I don't know if you know who Tom Barr is.

Booch: I've heard the name. But why don't you elucidate?

Humphrey: He was the top lawyer. He was the top lawyer for Cravath, Swaine and Moore on all of our legal stuff. A brilliant lawyer, and he's been in the news multiple times since. But in any event, he was there and there must've been about 30 people in the room. Basically, almost all lawyers, a couple of my staff were there. And so we started going through this thing, and I was going through it. And lots of questions by lots of people. And Tom Barr, as I say, didn't say a word. And after about the first couple of hours, I saw where we were, and I said, "Look. This is going to take a hell of a lot more than two or three hours." I said, "If you want to get through this, at this rate, we'll be all day. And what do you want to do?" And I said, "I can change my schedule and be here all day. Can you?" And we took a break, and they all came back, "Yep." So we decided to spend the day.

So I went through a whole lot of things. And long about two o'clock in the afternoon, Tom Barr asked a question. He said, "Watts, at nine o'clock you said this. About 11 o'clock you said that. Now, here you just said this. Those sound inconsistent to me." He said, "What's the story?" And I thought, this guy, he had a steel trap memory. Here he'd never heard about programming before and he had all his stuff together, and he knew exactly-- I couldn't believe it. It was just an extraordinary demonstration. I'll come back with another demonstration of something he did a little bit later. But it was just unbelievable that anybody could go through that and pick that up in that way. I thought there are some guys here that are just so extraordinary and that was an example. I've never-- I wrote a letter to *The New York Times*. People were questioning that he was getting $800 an hour, and there was a big plot going on. And so I wrote a letter to the *Times* about that and they published it. I was telling the story about this. I said, "This guy's extraordinary. Pay whatever you can get him for," and he really was. Okay.

Well, back in the timeline, another thing that came up was System Q. I mentioned I was working for George Kennard at this time, who was, you know, my boss. And Chuck Branscomb was Division President, George was the VP, Division VP, and I was the programming guy. And George asked us to put together a study of what would we do next with programming. And so I got the top lab directors, oh a whole bunch of guys, and very good people. And so we all went off and spent well over a week, and it must've been every bit of two weeks. We were holed up in some office-- some conference room somewhere. We had marketing people and the semiconductor people and the hardware people all come in and talk about future trends and what was going to happen. And I remember at one point during the discussion, we were talking about what future programming systems ought to be. And as you recall, the OS 360 was originally developed with an entry memory size of 16 kilobytes. We subsequently brought it up to 32 kilobytes. And when we had 128 kilobytes in a Model 65 and they actually got up to 256 that was considered [an] extraordinary amount of memory. And I remember in this meeting making a case that we-- the System Q. We ended up talking about our new system, the System Q, ought to have a minimum memory of one megabyte. And they thought I was crazy. But anyway, so I was, basically, probably the least knowledgeable software guy in the room. But I was sure right on that. I way understated what we would eventually need.

But we had no concept of how big this stuff would get. But in any event, so we went through System Q and what it ought to be. And it was really a marvelous concept of a highly-interactive system, communication- based, virtual memory and that sort of thing. It was communications, all the elements we had updated, the compatibility and file interchange and all that stuff. We couldn't visualize the Internet, but it was the system you'd need for the Internet. It was a marvelous system. But we never could get to it. And they didn't leave me in place long enough. Along about there they had another division re-organization, and I was made a VP. And I was given the architecture group, which had Dick Case. This is the group that Amdahl had run. And so I was getting that as well as the programming crowd. I had all the programming labs, and I had the hardware computer architecture stuff. Don Gavis ran the 360 stuff for

me. Dick Bevier had all of the programming labs for me, and we had a couple of other system managers. Jim Frame had the small systems.

Booch: Can I ask a question, along the way, here?

Humphrey: Um-hum.

Booch: So where were you getting the people who had the skills to be programmers? Were you in touch as a corporation with the universities to help encourage this? Was IBM doing its own training internally? And where were the programmers coming from?

Humphrey: We were typically hiring the best people we could get out of college. You couldn't get good people trained in what we needed, so we, basically, anybody we hired, if they passed various tests and seemed to be smart, we put them through training. And so we were, basically, training all of our own people. And so they would come back, and they would start training. And they'd go through and learn programming the way IBM did it. They'd learn the programming languages and work with our systems. So we, basically, had an education program. We trained people. And it was expensive and a problem. But early on, we weren't getting-- we were getting marvelous people.

We got top graduates from the best schools. But there wasn't much in the way of really good training. I mean, they'd come out knowing what computers were and that sort of thing. And they had some knowledge of programming, many of them. That got better and better as we went along. But we, basically, had a training program we put everybody through.

The Virtual Memory Decision

But in any event, now let's-- yeah, so I was made VP, and I had the architecture stuff. And as I said, Don Gavis had the OS 360 work. And he came to me, I think it was in about 1969, '68, '69-- yeah, it was '69. And said, "We've really got to go to virtual memory." And remember, the TSS was killed and the 360 didn't have virtual memory. And we had that big battle with Amdahl that… “just add more memory.”

But the programmers had concluded that virtual memory was probably the only way to go. We just had to do it, get out of the constraints of the physical memory. And just about that time-- at the same time IBM was developing a newer version of 360 called the 370. We'd been out there for a while with the 360 systems, and people were beginning to catch up with us, in terms of performance and that sort of thing. And there was quite a lot of competition. It was still pretty vicious. And so they were coming up with upgrades and higher performance hardware and that sort of thing. And they decided to call it System 370. We were working on 370 development work. And so the recommendation was that we switch System 370 to be a virtual memory system.

Well that was a radical change. Some hardware guys were happy to do it, but a bunch of them weren't. We had a bunch of microcode machines, where were fairly easy to switch. But the hard-wired, bigger systems were a much tougher problem. And right about that time, they had a re-organization. And a new Division President was brought in. And I'll come back. There's a story about that that I want to tell. And guess who was made the Division President? It was Bob Evans, the guy I'd had two previous battles with. On both the timesharing stuff and on the FAA thing, and I won both of them. And so Bob and I weren't on

the best of terms. And he'd gone down to run the Federal Systems Division, where they programmed the FAA system, if you remember. And he was brought back as Division President.

And so one of the first things we did was to go in to Bob, I did, with my programming team and the architects. The architects agreed that we ought to move to virtual memory. So we went to Bob. And of course, we were fighting with the hardware guys. And so Bob looked at it and went through the story. I mean, I have tremendous admiration for the way he was able to take a multibillion-dollar decision and made it in a day. I mean, he went through this, he looked at all the choices. And he said, "You're right. We'll do it." And here he'd been opposed to it. He and Amdahl and et cetera, but he went through the logic and what the guys were talking about. And he was a very sharp guy. I had differences with him, but that was-- he knew what he was doing. He made that call and he was clearly right. So that's how we put virtual memory in 370. Bob made the call and it was, basically, Don Gavis that turned us around. Now, there's another story here. Oh yeah, well, also, at about the same time, CDC had sued IBM, antitrust lawsuit, big antitrust lawsuit. And I was…

Booch: Who was in charge of CDC at this time? Who were the folks you were negotiating with or fighting with at this time?

Humphrey: I really don't remember who was running CDC at the time. I think it was still the original management team, but I'm not sure. So I really wasn't deeply involved. I was really hunkered down in the development business. But in any event, I had to get deposed. Sound still okay?

Booch: Yes, very well.

Humphrey: Okay. And so they started these depositions. And I ended up, I think it was, like, six weeks of testimony. Just extraordinary. Like, 1,600 pages of stuff. And the lawyer for CDC would start off each day with this big pile of mail. And he'd refer to a piece of mail that somebody wrote and I was on the address copy list. And he'd say, "Here's what this guy said. What does that mean?" And if you remember, I didn't get any mail. I basically said, "No mail." And so these guys were really kind of upset, because I didn't see it. And if you want to know what he meant, you're going to have to ask him." And so they basically got nothing out of me with six weeks of this stuff. It was an enormous waste of time. I will say, however, that there was one thing that came up in that discovery.

I subsequently learned that came up, it was a letter that I had written later. No. I'm a little out of line here. I'll have to come back to that. Let's see if I've got that down here somewhere. Yeah, it's in here. Let me-- boy there are lots of things. This is the compatibility letter, yeah. Okay. So in any event, we did the lawsuit stuff and the depositions. But it was, sort of, a dud. But it was amazing the amount of material they had. They must've gotten everything. There was another incident that happened in there. And one of the questions that you might say is why, when I had been so right on all of this key stuff and Evans was wrong, why was Evans promoted instead of me to be the Division President?

Skiing with Tom Watson

And it was funny, because I'm convinced there's an incident that happened there that, kind of, soured my relationship. And it was in 1968, Tom Watson-- or '69. Tom Watson was going to retire at age 60. And so Vin Learson was going to take over as CEO. And Tom, he hadn't announced he was leaving, but he was

obviously planning on it and doing it shortly. And I think he was, basically, trying to figure out what kind of structure he wanted to leave behind him. And so he decided to invite all the top development people to one of two events. We would all start off in Denver, and a bunch of us would go up skiing to Aspen with him. Another group would go with Learson over to a big resort area, the non-skiers would go over there. So that's what we were going to do. And I had been skiing. I'd taken up skiing myself, taught the kids to ski and we were skiing up around New England. And we had a few lessons and stuff, and I thought I was pretty good, but I wasn't. I mean, I thought I was a whole lot better skier than I was. And they asked who wanted to ski with Tom. And I thought I was up to it, so I did. I signed up to ski with Tom. And there were going to be a few of us doing that, Dean McKay and a few others. And it turned out I was way over my head.

Booch: Didn't know that Tom was a good skier.

Humphrey: Tom was good and his family was there. And so I was tired. I couldn't handle it. I wasn't that good at it. And I was terribly embarrassed. And I remember, one day, skiing down. I mean, basically, I backed out. I, sort of apologized and went off and joined another group. So I backed out fairly soon, I think, the first day. But the end of one of the days of skiing, as I was going down on one trail, way off on one side. I saw that Tom Watson had actually had a fall and was, sort of, in a drift there, kind of, struggling. And I was so embarrassed I didn't stop and go help. And he saw me. I'm sure he saw me as I went on down. And I think he just felt that was totally ungentlemanly. And I think that was a real negative. But while he never did anything to stop me, I don't think he pushed for me for the Division President job.

And so it was funny. That was always a problem, because, basically, that's when my career goes stop. Up to that point, I had been on this rocket ride up, and all of a sudden, I wasn't. And what's interesting was, it was a terrible shock to me to realize that my career moving up to be a top executive and a president and CEO of the company, which I thought was in the cards. I later learned it was. I was on the list to go there to the top. And it took me a number of years to realize that was probably the luckiest incident that ever happened to me. Because I later realized that I would never have done what I'm doing now. And that I really wouldn't have enjoyed being one of the top executives. I've really gotten to know what they do. I learned that a lot later. And it was not a job that I would've enjoyed. And so it was, kind of, hard to face that when you want to be king and discover that you're not really set to be king. You want to be something else.

Booch: It sounds like the world had other plans for you.

Humphrey: Yes, indeed. But it was the dumbest thing I ever did. But in any event, so we had the division reorganization. Bob Evans was brought up. He called me in to meet with him. He said there were two people that were in line to be President and CEO of the company. One was me and the other was Jack Kuehler. Bob himself apparently wasn't on the list. But, for me to continue moving on up, I had to run a lab, and I had to get some experience doing that.

Booch: Um-hum.

The Endicott Lab

Humphrey: I agreed to take the lab in Endicott. He said I would no longer be a division vice president, because, you know, "We've abolished those jobs. You could stay here and in that job. I don't know what you'll do," but et cetera. And so he was quite direct on what I ought to do. So I agreed to take the Endicott job. And so a bunch of things, I got a few stories about what happened there when I ran it. It was up in Glendale. It was IBM's oldest laboratory. It had been there forever. A very nice guy, Jim Troy, had been the Lab Director and was retiring. But he'd been in place forever. And when I got there, I discovered they had, like, seven layers of management and an enormous array of staff people.

So I decided that we had to re-organize the place. We had an awful lot of work to do. I had all the intermediate systems, the printers, the OCR/MCR stuff. It was a very important lab and we had several billion dollars worth of products. The mid-range computers were an extremely big market. It was billions and billions of dollars in hardware. So that's what we did, and it was about 2,000 engineers and we were working with a big manufacturing plant there in Endicott. And so I had the-- I put together a committee of the line managers, the people developing the products. And I said, "What I want you to do is to go through the list of all the departments in the lab. And tell me which ones you need to do your job." And so they did. And they went down this list. And there were a whole mess of them that they indicated that they didn't know what-- "Why have we got those? What are they doing? We don't need them." And so I started down the list of them to have each of those other departments to justify why are you here? What is your job? What makes your work essential to IBM?

And we eliminated-- I mean, we kept the technology stuff. No one said they needed the tech stuff, but I said they did. So Sy Tunis was running that and he kept it. So we kept that. He was our next-door neighbor, as a matter of fact, when we lived up in Endicott. And uh. Sy Tunis was his name, marvelous guy. But we went through that and identified 200 managers. It was hard to believe. And so we re-organized and, basically, moved all of them into real jobs. Most of the managers that were excess, we made non-managers. There were lots of managers. We had a manager that had two people reporting to them and things like that. So the hierarchy was seven to eight layers of management under me in a lab of 2,000 people. It made no sense, at all. So we got it all the way down to five levels, got it tightened up. And what's amazing was morale improved dramatically. People were in real jobs, even though they didn't have quite the same prestige they had before.

And it was amazing we were able to do that. And we did, basically, requiring that people justify why they're there and how they're supporting the products we're developing. That was why we were there. So it was really a hell of a good reorganization and it worked very well. Also, there were-- had a bunch of battles with Bob Evans. Bob had a habit of going out to various labs and looking at what ad tech was doing and if somebody had a wild idea, they'd bring it to him. And somebody had an idea for a disk printer, a little printer that-- where you have little flexible plastic disks. And you'd hammer the disk onto the paper. And it was-- he showed us it would be real cheap, easy to make, et cetera. And so Bob essentially directed us to develop a disk printer. And we disagreed with it. The printer guys didn't want to do it. The printer guys were really difficult on that. The manager was a guy that I had a lot of battles with, who worked for me. And the basic problem was that he had expected to be the Lab Director, instead of me. And he'd been with the company a lot longer and that sort of thing. And so we did not start off well. And we never did work well.

As a matter of fact, he was-- turned out to be a real problem later. So we had a big battle with the disk printer. They were never able to get it to work, and they tried all kinds of stuff. It was an enormous expense and big failure. A little bit later, as I said, I had this battle with the printer manager. And I ended up pulling him out, and I wanted to reassign him to run the OCR/MICR stuff, but he wouldn’t have any

part of it, and we got into a big flap with that. And so, I had a bunch of problems with people that I was having difficulty with and a lot of them were old timers that had a lot of background that we had some real trouble with. It was a tough environment to work in because anybody could go open door if they disagreed and any of that sort of thing. You couldn't move people without all kinds of justification. It was a very difficult way to do things. But in any event, we also had the OCR/MICR Group, optical recognition, which was in Rochester, but they were actually working for me in Endicott and Pat Beebe ran that group and I'll come back to him because Pat Beebe, the reason I mention him; fairly early on, I had worked for him— Remember when I was telling you the brokerage studies work and that sort of stuff?

Booch: Yes.

Why RCA Failed

Humphrey: Well, Pat Beebe was my boss then. Actually, he was my boss' boss. All of a sudden, I was his boss and boy, that didn't sit well with him at all.

Booch: I can imagine.

Humphrey: Well, that was the case with several people, that basically here was a guy from lower down that came in. Of course, I'd been running programming. So, I had been in all these big jobs. So, I rocketed past all these people who felt that they had earned their keep and I hadn't and so, I had all kinds of problems with them. So, Pat ended actually resigning from IBM. He quit and I had arguments with him when I worked for him because he [had ideas that] turned out not to make a whole lot of sense and basically said you couldn't make a general purpose programming system, operating system. He said they all had to be special purpose. He was a special systems guy. So, he was just plain wrong on some of this stuff. Again, one of these people that had a conviction and essentially made up facts to support convictions instead of testing convictions against fact.

Booch: And, he had grown up primarily from a hardware perspective, isn't that correct, where there was the legacy of building specialized machines.

Humphrey: Well, he actually was the guy who ran the SABRE System development. And so, he had that background - the special systems stuff and how complicated they were and all that sort of thing. He came from that community; a hardware guy, but also through the SABRE stuff. So, he had a pretty good extensive background and he was no dummy. He knew how complex these big systems were. So, that was an issue. He had been in charge of the early development of that and that brings up another point. I ought to mention the SABRE at some point. I don't remember when that was. The SABRE thing, I think, was a little bit later - about the time of the PC story. So yes, on the SABRE performance. I'm dropping in a lot of things here. Pardon me.

Booch: This is great. There's so much richness and history here.

Humphrey: I learned a lot. So in any event, Pat quit. Do you remember Orville Wright was my marketing manager when I ran TSS?

Booch: Yes.

Humphrey: Orville Wright also quit and they both went to work for RCA. There's an amazing RCA story here because Pat, when he quit, he said, "I'm going to RCA and we're going to clean your clock" and was really aggressive on that. He had taken the job as the VP of development. Orville Wright was head of their computer operations. So, he took it over at RCA. He had a big job. What was interesting was RCA had come up with, as I said, the Spectra 70, which they had gotten going, but they were replacing the Spectra 70 with a new line. What they had done, which I learned later, they made some just god-awful mistakes in a marketing and policy basis because they established a series of long-term leases to sell their machines and they were actually pricing them with fairly long lives.

So, they were able to compete pretty aggressively with IBM, and they were getting business. They were doing reasonably well. And so, when IBM came out with a 370, they had a problem because they were developing a new replacement system also with higher performance technology and that's why Pat was brought in - Pat Beebe and Orville Wright - to go in and fix that and bring this new system out, but the new system was delayed, and it was pretty seriously delayed. And so, what they decided to do, Pat and Orville, they decided that instead of coming up with a new machine, they would fake it. So, they would say, "Okay. Here's a new machine." They would take the old Spectra 70 machines, bring them back and repaint them and do a little bit of clean up to them and ship them back out as a new model at a much lower price. And so, it was kind of fakery.

And so, they did that and what they didn't realize was that all of their long-term leases had granted an out. You had to pay a termination charge to get out of the lease unless you were buying another GE machine like getting another Spectra machine. So, it turned out that all of their customers could now take the machines they had, essentially trade them in and get one of these newer models at a much lower price without paying a nickel. So, the new RCA operation; all of a sudden, they were running their factories like crazy, retouching all these machines and cutting the hell out of their revenue.

And so, these guys went down to the RCA board meeting one day to decide what to do about this and overnight, they wiped out the RCA computer division. The board just said, "No. It's gone." Oh, and it became a money pit. They were into the hole for like $100 million in no time. And so, that was sort of the dead end for those guys. Orville came out well. I don't remember what Pat did. I heard him of running some smaller company somewhere. But in any event, we also had another big flap. Learson now had taken over for Tom Watson. Tom Watson had retired. I was still at Endicott. When Learson took over, this was about 1970-1971, they had to cut people. And so, Learson basically came up with this thing that you have a requirement to cut out so many people. They levied a number on each lab. I got a number. I forgot what it was - 70 or 80 people I had to get rid of, but I had to follow standard IBM ground rules. I could only fire people for cause. It was ludicrous. I mean I had to go around and figure out some way to fire good people that were doing good work. I totally objected. I had big battles with my immediate management and so I got in real hot water over that again.

My boss at that time was still Bob Evans. He reported to a guy named Spike Beitzel who was the group executive and I really got into a terrible flap over that thing because I just— It was unfair. We were being dishonest with these people. If we were going to fire them, let's come up with the layoff number and they were unwilling to do that. So, we had to somehow make it up. Learson was driving that. While I thought he was an extraordinary and marvelous man, he sure had this hard edge to him on this. This was a toughie. But, that was a real problem. So, I got in a real battle over that, and then had another re-organization - IBM kept re-organizing - and decided they were going to put a whole bunch of things together, restructure it. One of the big debates we had at the time, and this will come up a little bit later; we had a big meeting with Bob Evans and all of his lab directors. He'd had these lab director meetings

periodically and we had them in everybody's labs. I remember we had them in Europe and all over the place. So, we would all go to the lab directors meetings.

At this one meeting, Bob wanted to talk about how should we structure the division in terms of the way we lay out the logic for the organization. The option basically that he was proposing and pretty much everybody else wanted was we will organize around computer systems. We have a large systems group and so, each of the systems would be the center and they would each have their own programming group and that sort of thing. So essentially, we would fragment the whole programming community. That was sort of what had been done before, but it hadn't been broken up. And so, they wanted to really now essentially break it up so everybody had their own programmers. All of the hardware system managers were now the system managers and they would have their own hardware and software. The file people would get the file software. So, that's what they were going to do. I objected. I said that makes no sense at all. I said that what you really need to do is to have an operating system focus instead.

So, focus on what the users are doing and do it that way. Well, the whole hardware community would have none of that. So, they basically decided to re-organize that way. So, the whole programming community was essentially splintered and that happened, as I said, right about the time of the re-organization when I went to Endicott in January 1970. They later asked me now to go down to corporate staff and take a job as director of policy development. This was about in 1972.

Booch: If I may ask, around that time when you were still as the director of programming, what was your span of control because you told me these stories of all these groups you're working with. What was sort of right below you that was the structure below you that then reported to those people reported to? What was your span of control?

Humphrey: Oh, okay. I ran programming from 1966 to 1970 and had a whole lot of people under me. I had Fritz Trapnell, who was the director over the OS 360, later became Don Gavis and Fritz, by the way, went over to England and ran the Hursley lab for a while.

Booch: Got it.

Humphrey: So, Fritz reported to me. Jim Frame had the intermediate systems, DOS and that sort of stuff. We had Dick Bevier and all the labs reported to him. So, the lab managers, who were different than the system managers, by the way; so, there was a Poughkeepsie lab manager and there was a San Jose lab manager and there were other managers as well. Now, some of the managers; for instance, the people over the data management work and the sorts and all the, they reported to their programming lab managers.

But, the big operating systems, the large and intermediate and the small, they reported directly to me and Dick Bevier had the labs. I also had my own very small staff. So, I had about five people. I didn't have a big crowd. Dick Bevier had quite a list of lab directors, although the European people were all kind of pulled together in one group under By Havens. But other than that, they were pretty much all under Dick.

Booch: Okay. That helps.

Humphrey: So now, from 1970 to 1972, while I was in Endicott, running the Endicott lab; now, I was no longer the programming director. I was running the Endicott laboratory.

Booch: Right.

Humphrey: So, I moved out of that. A fellow named, who had worked for me; they were looking for who to have run programming as essentially a staff job. You remember now the programmers were all reporting to their local laboratory management chain. When I went to the Endicott lab, all the programmers were doing that as well. So, they had like the Poughkeepsie lab and all of that. They were reporting to the lab directors, whoever was there where they were working. So, I had actually recommended that one of the staff people that I had in my group, a fellow named Ted Climas; that he take over as the director of programming. So, he actually got the job. He was very good at it. He was a great big, tall guy. So, Ted Climas actually ran that. While I was in the Glendale lab as the Endicott lab director, they decided- Beitzel decided that they needed a new way to attack small systems.

IBM was being aggressively attacked at the very low end. It wasn't PCs yet, but small computers - DEC and others. And so, the hardware guys had concluded they really needed to make a small system that was not 360 compatible. They were going to break the line and go off on their own - the System 38 I think they ended up with. I was irate. I thought this was crazy. So, I wrote a blazing letter to Spike Beitzel about compatibility and what we would do and I got Dick Case, who had worked for me as the 360 system architect, to bless it. He went over it and commented on it. I didn't anticipate the Internet, but I basically said that compatibility, interchangeability, that there were various levels of it and you really had to have the ability to move data and programs back and forth, that people were going to do this stuff dynamically and we had to be able to interchange between all these systems, and that going off with a completely different system was a terrible mistake.

That letter actually was— I never kept copies of any of it. I'm not a pack rat, so I don't have much of that stuff, but it showed up later and I don't know if I could find it… I think I could. It's stuck in the depositions and ended up in somebody's book. I have a search thing. I may be able to find it here. Let me find it. But in any event, it created no affect at all, but in any event— How do you spell "compatibility" - A-B— Compat— P-A-T-A-B-I-L?

W: I-B.

Humphrey: I-B?

Booch: She's not only a wonderful bride, but she's also your spellchecker.

Humphrey: Okay. Yes. Let me see here. Okay. I can get that. I'll have to pull it out of the reference. I got it.

Booch: Got it?

Humphrey: Delamarche. Actually, it's one of the things that I may do. I've been debating about what to do with my enormous pile of stuff. I've got a collection of all the papers I've written, talks that I've given, my books; stuff like this I can pull out.

Booch: The Computer History Museum is a great place for that. There's a very vibrant group that's doing the software collection and exactly those kinds of archives are the things that they're keeping. With John Backus, we've collected a lot of his early papers and some of his source code and the like and this would just be a treasure trove.

Humphrey: Well, I've got that. I keep notebooks, but unfortunately they're just my scribbles. That's sort of what I do, but occasionally, I note meetings in them and I'll jot down ideas and stuff like that. They may be unintelligible, but I have 58 notebooks - I have all of them - basically starting principally from when I got to SEI. I don't think I've got any from earlier than that.

Booch: Wow. What an amazing treasure trove. It may be indecipherable now to anybody but you, but there are generations of people who I would imagine would love to pore over those kinds of things.

Humphrey: Well, I've got them. I can send them out to them. I'll pull out some of this stuff and I've got a bunch of files and records and stuff like that. I'll send it out to them at some point. When you chat with them, you might ask where they want me to send it because I can send it to some office out there or something, but it'll be a while. I'll pull it together. It'll come in pieces.

Booch: No worries, but yes, I will blaze the path to make that happen.

Humphrey: I'd be happy to do it. So, that letter I have. I could pull out a copy of it from this book and just include it in the list. But, it's a great letter. If I get a chance, I'll pull it out. I can read some of it to you if you'd like. (See also The Compatibility Letter, day 2 PM.)

Booch: To give you an example of what's collected, and this is a little off the path here, we just announced; the museum just announced we got a copy from Apple of their original business plan for the Macintosh. So, those kinds of things, they're great to have.

Corporate Policy

Humphrey: Oh, no kidding. Yes. Okay. So now, as I said, the compatibility letter and they decided to do— After the forced attrition and all that stuff, they had another re-organization. It was much along the lines of what Bob Evans was talking about. They put in a location manager over the lab and manufacturing in Glendale and they did the same in each of the locations. So, they completely restructured it, and I ended up taking the job on corporate staff as Director of Policy Development.

So, that turned out to be a fascinating experience, but it lasted a long time. I was there from 1972 to 1979. While I was there, we got involved in lots of stuff. I remember I got called in sort of - I'm trying to remember where it was. It was an issue on the performance of the SABRE System. This was not connected to my policy stuff, but I was a technical participant on the taskforce and this is earlier. I think it may have been in the earlier corporate job, but it's in my notes here, probably in 1963. And so, I remember we went over to the SABRE lab. This was after Pat Beebe had moved on, but somebody was running SABRE. It was a big American Airlines reservation system and it was before SABRE had really gotten going. They were still doing the development work.

And so, we went out and had a review of it and one of the questions was the data rates. The question was what kind of data rate are you planning for, and they were talking about ten transactions per second. I said, “Well, how would you handle a thousand?” They said, “Oh, we’re within the specification for a maximum of ten.” I said, “No. Forget the specs. I said what would happen if you had a thousand?” “Well, the system would crash.” I said, “You need to rethink that.” That’s basically what the taskforce did and it turned out they had some kind of handshaking system that would break down with larger data rates

Booch: So, at least they knew how it would fail, how it would fall apart. That’s a good sign.

Humphrey: They did know that. They did redo it, but it was amazing. I think I heard that the SABRE workload was running at about 10,000 transactions a second. People couldn’t think big. Like I said, it was true of memories. It was true on the performance of chips and everything. Okay. Well, anyway, back to my corporate job; I moved on to corporate staff in 1972, Director of Policy Development and it turned out a big part of my job because the lawsuits were going on and they wanted someone in corporate that would be involved from a business side. I had an MBA. And so, they felt that that was a good connection. I had a lot of background with hardware and software.

Booch: Besides, you were a machine gunner in the war. So, you had a lot of experience.

Humphrey: Yeah, there were times I could have used a machine gun. So, I went down there. Actually, the head lawyer— No-no. The head lawyer - I’ve gone blank on his name - he was attorney general under Lyndon Johnson. His name will come back, but a marvelous guy. I would go in and meet with him often. He was a wonderful guy to deal with. The lawyers were great to work with as a matter of fact.

Booch: Ramsey Clark - does that sound familiar?

Humphrey: Oh, no. No, that wasn’t him. It was Nick Katzenbach. The Chair he used as a Cabinet Secretary was the chair he had in his office. It was his chair. So, he was there in this big leather plush chair. He was a wonderful guy. And so, the lawyers at IBM really understood one thing very well. They understood that their job was not to make decisions. It was to give advice. The businessmen made decisions. I was a businessman and they were giving advice. But boy, when you listened to them, they were really good.

And so, he was a marvelous guy to work with. A lot of lawyers get into trouble that way. They think their job is to make decisions and unless you’re in a trial because there they make decisions of course. Okay. So, I was down there. Oh, one thing I forgot to mention that come up. Yes, I remember I talked about— Let me get this in. I had talked about this would have been at about 1967-1968 after we had all these discussions with the lawyers and I described what programming was and Tom Barr and all that sort of thing. I got a call one day from Tom Barr’s office. This is like on a Thursday afternoon or a Friday morning. They said Tom Barr has to be in court Monday morning on a lawsuit about computer sorting.

Booch: On computer sorts? Source?

Humphrey: Yes, sorting programs.

Booch: Oh, sorting. Oh, my goodness. Okay.

Humphrey: Because apparently, IBM had been sued by some company that had a sort program and they were after us on this program and they were out for a bunch of money. And so, the case was in court and Tom Barr had to be there Monday. I don't know whether sort was the basic lawsuit or just it came up all of a sudden. They had this witness and he was an expert on sorting. That's what it was. Tom Barr had to cross-examine a witness who was an expert on sorts on Monday morning. So, he had all weekend to prepare.

They called on me like on Friday. "What can you do about that?" I said, "Well I can get a hold of our guys." We had some sort experts in Poughkeepsie. I got a hold of them. Of course, when the lawyers called, that was the top priority. So, they said, "Okay. We can do it. We can meet with them Saturday and Sunday in Poughkeepsie." So, I got their names and numbers and I got back to Tom's office and said, "This is the guy you can talk to. Get a hold of him. Get in touch and go ahead." And so, he did. Monday in court, Tom Barr cross examined this expert in computer sorting and took him apart technically. He actually asked him a whole bunch of questions about sorting the guy couldn't answer. He destroyed the witness and I couldn't believe it. I heard that he had done a marvelous job, and it sounded like he had been doing sorting work all his life. Tom Barr was just an extraordinary person. He had an ability to pull all this stuff together. So, it was just amazing to me what he did.

Contract Problems

So in any event, now I'm in the corporate office, in the corporate staff in 1972. I reported to Dean Phipers I think his name was and I was sort of the next layer down from the president, chairman - all of those. The job I was in, basically all policy changes went through me. The kind of things that came up; one was the new purchase and lease contracts. IBM had had a battle with collecting money from a small jewelry company in Rhode Island. And so, we were after them. We were just collecting the payments they owed us. They had trouble with the system and they were complaining that IBM had failed and we had screwed them up. The company was in trouble and they weren't going to pay and this sort of thing. So, IBM would take them to court just to pay their bill.

And so, they got a smart lawyer from New York who basically looked at this stuff and decided to file an antitrust lawsuit against IBM. He was going through all this stuff and as part of that, it turned out the old IBM purchase and lease contracts, and the lease contracts turned out to be specifically for hardware and it would exclude everything else. So, the contracts didn't apply to software. It had all been excluded. And so, all the warranties and guarantees and limitations, because we had all kinds of warranty limitations in there. And so, the path these guys were taking was that all those warranty limitations and everything apply only to the hardware. They don't apply to the software because the software is excluded from the contract and as a consequence, any promises made by the salesmen or anybody else were legally binding.

In fact, we lost the lawsuit on those grounds. And so the problem was, that IBM discovered, and this came out in a Board meeting, when I was sitting there. I was sitting right behind John Opel, who was now President of the Company, or CEO of the Company. I was sitting right behind him. He was at the table, around where the top executives sit, and the staff guys were- on this particular meeting, I was in for that, so I was sitting there right behind John. And this came up. This is why we lost the lawsuit. And John Opel turned around to me, and said, "Watts, will you fix that? You know what we had to do, we had to rewrite every IBM contract, every purchase contract, every lease contract, every maintenance contract.

We had programming contracts, we had to go through everything, and what we realized was—about liabilities. You can't limit liabilities without actually making commitments, so if you're going to limit your liability, you have to make sure that you are in fact committing to what you're limiting about. And so that was what we had to do. So we did, in fact, rewrite every purchase and lease contract for IBM worldwide.

Booch: That would have been thousands of contracts.

Humphrey: Oh yeah, it was an enormous number, and so we put it together-- I was the final source. We'd go through, we completely restructured it. We included warranties for software. It was principally that the software will be there. Fundamentally, it's the basic limitation they've got on all the stuff right now. We had no limitation that it would run, the limitation was that the code was there and not what it would do. It was present on the media that we delivered it on, and that was basically it. Now it's obviously evolved a bit since then, but we had to put that in place, and we had to put basic warranties in place on that software, which we did, and they were pretty rudimentary, but we did it. I think those are still pretty much what they've got, so I'm the guy responsible for all that stuff.

Booch: It's all your fault.

Humphrey: Yeah, so we put those together, and it was an enormous risk, if we hadn't done that, we had to do it, and the problem we had, of course, what you- a lot of the contracts we had, the IBM lease contract was a contract that lived in perpetuity, you couldn't get rid of the IBM lease contract unless the customer was willing to, because it had a renewal clause right in it. So the customer always had the option to renew the old lease contract.

The purchase was different, of course. And so we basically, we did the complete redesign. We had to make the contract attractive to the customer. We had to be very sure it wasn't forcing anything on them that they would be upset about. We went through it very carefully and I remember translating the contracts they actually had a different contract in England than in the US, because of course laws are quite different there.

And I remember one of the big problems we had was with the finance community, surprisingly enough. The lawyers we didn't have too much trouble with. We'd go through, work out all the language with them, but the finance people were real sticklers on negotiating all these contracts. And I remember the British people came over to negotiate the final language on their contracts, and so I was involved in all these things. I was really, this was a hell of a job.

And I remember, we were going through a contract, and it started to snow. And we were working on it, and we weren't able to get all the way through, and so Friday afternoon, everybody sort of had to break out. And I said, "Well we'll meet again in the morning." This was Saturday morning, and the finance guys said, "No, we can't meet tomorrow." And I said, "Fine, if you can't be there, then you can't object, and we'll go ahead without you." And so we did.

That's sort of the way we did all of this negotiation. We did it on a fast track, and if people weren't able to stay till the meetings ended, until 2 in the morning or whatever it took, we'd go till we were done, and the people who had the guts to stick around, got to speak. But we basically got it all done and then, when the contracts were all completed and ready to go, we couldn't introduce them because the administrative

community hadn't fixed the programs yet, so the administrative programs to actually administer the contracts had to get rewritten, and we couldn't get a priority on that to get it done. But we finally got it done.

But we got the contracts out, and we replaced them worldwide, and we ended up without a single lawsuit, which is kind of amazing. It went across smoothly, everybody took them. It was-- the new contracts were the only way you could buy the new equipment, any new machines, hardware or software. One of the big issues we had was that every time you go to negotiate a contract, the customer's lawyers always want to change things. And so I started to get these phone calls. And so we finally concluded that if the language-- you could change any language you want, as long as you don't change the meaning. So we gave our marketing lawyers the freedom to change words, as long as it didn't legally affect what was meant, and we were very precise on that. The lawyers had to make damn sure that, from a legal perspective, the meaning was identical. We gave them that flexibility, and all of a sudden, all the questions went away. Customers and lawyers were happy if they could go in and change half a dozen words. All of a sudden, they had done their job, and so we got, we replaced all the contracts without a single lawsuit or a single problem. I was astounded. It worked like a dream.

Booch: Amazing.

Humphrey: Yeah, so it can be done. The customers were perfectly pragmatic. They weren't going to fight us over this nonsense as long as we dealt with them in a realistic way, and we did. It worked fine. One of the things that, now let's see. Yeah, one of the things that I was going to talk about here, was the IBM business system. One of the guys, who was a wonderful guy, he was the guy who was in charge and handled the pricing for us on the FAA earlier. His name was Hilary Faw. He had been Director of Policy Development earlier, the job I now had. And he was a respected old timer, and I think he was still there in 1972, or he was just about to retire, or was retiring, I don't know. But he had actually gotten people working at what he called the IBM Business System, and as a matter of fact, I remember, I digress a little bit before I forget it, earlier on, when-- Well let me talk about the system and come back. Let me make a-- I've got another one I've got to insert, which was a meeting I had, which would have been back in '69.

Booch: Watts, I have to say, I am stunned on two levels. First, just the rich tapestry of things with which you were involved, and second, your ability to recollect all of these. It's quite impressive.

The IBM Business System

Humphrey: Well they pop up at random here, but I've got to go back to this meeting. But in any event, so Hilary put together this thing, what the IBM Business System was and the structure of the Business System. It was basically around the whole lease framework, that when you lease a new machine, you get more revenue, so the question was lease life, and all the rest of it. The whole Business System, how it worked and the feedback systems and the nature of the lease business, and how effective it is, was extraordinary.

And that's how IBM had run, and how they've been so enormously successful financially, because every time you lease a machine, all your marketing effort now is out to sell more stuff, and upgrade the machines you've got. So the stuff you've got out there continues to gather revenues, so your revenue base is just constantly growing. It was marvelous. It was a great system, and you had to, sort of keep the

machines up to date and that sort of thing, I think, but it seemed to- just the magic, make an extraordinary amount of money.

Booch: And were IBM's competitors doing the same thing, the seven dwarfs around that time, did they have a similar strategy?

Humphrey: They all had the same basic strategy. What had happened, however, and IBM had had a previous antitrust lawsuit which was settled in 1956. Remember, I joined the company in '59. The previous antitrust lawsuit was settled in 1956, essentially over Tom Watson Senior's dead body. He really objected strenuously, he fought it like a tiger, his son had concluded that he had to do it, and so they ended up, they had agreed, they signed an antitrust agreement. They never went to court. They signed a Consent Decree, and Tom Watson Senior died, like, that year, so it was '56, and in that Consent Decree, IBM agreed that it would sell machines as well as lease them, and it would sell them at some presumed life, so there was some ratio of lease to purchase price that was, the Consent Decree put some constraints on it, so we couldn't put the purchase price way out of range, so purchase *was* an economic reality. When we announced a new machine, the antitrust lawyers would examine the pricing.

And we also had to permit other people to maintain our purchased equipment. And to do that, that meant that we had to make maintenance parts available and we had to be able to break out the maintenance price, so they could price it themselves. We had to price it separately, and we had to price spare parts separately, and as a matter of fact, we had to make some maintenance documentation publicly available. Well all of this was sticking a spear right into the IBM Business System. So that Consent Decree basically opened the door for all the plug compatible guys that came in later. And that was the real threat to the IBM business. I mean, it fundamentally changed it, but it took a long time.

Other Legal Issues

And so from '56 to the early '70s, was how long it took for the plug compatible community to catch on that they could do this, and a bunch of them came up as a result of that, and I hit lots of them when I was on the policy job, and one of the ones I remember was a lawsuit we got and I got this phone call. I mean, I had a policy representative in every region of the company, and both around the US and Europe, all around the world, everywhere, so if something came up, people would call me. I remember, I got one call from New York City, a guy called to say that one of his salesman had been out on a date on Saturday-- No, it was on a Friday. It was late in the week, and I got the call on a Saturday, or I don't know when I got it, but at this call, that salesman had been on a date with this young lady, it turned out her boss ran the whole company.

Booch: I'm sorry, you broke up just a minute. He went out on a date with this young girl, and-- and you dropped out for a second.

Humphrey: Now what happened was that this salesman had gone out on a date--

Booch: Yeah, I know you hadn't, you were married. <Laughs>

Humphrey: <inaudible> But this guy called me to explain that. This was a Branch Manager called me, or it was-- that's what it was, it was the Policy Representative in the Eastern Region, called me with this

story, and I guess he must have had my home number of something, but anyway, he got hold of me, and told me that, "Oh yeah, this lady mentioned that, "Oh, we're going to sue you Monday." <Laughs> So he called me, and I said, "What's going on?" And he told me who the company was, so I called our lawyers right away, and said, "Here's what I just learned,"

And so they got somebody out in this President's Office Monday morning. And we were able to stop the lawsuit, but what it turned out was, it was a little company that had gone into business with buying and selling features for IBM equipment. And the lawyer told me, when he went out to meet with this guy, he said, parked in the parking lot were you know, Cadillacs, a Mercedes, he said, "These people were doing well. This was not a low-price operation." They were doing extraordinarily well, because what they had discovered was, that the way IBM priced features had no relationship to cost. And if you really got in and understood what the maintenance stuff was all about, you understood how the feature was actually put in place. For instance, the feature to upgrade the Model 30 computer, from 32 to 64K. The computer was basically built with 64K in it. And when you purchased a 32K machine, they had a wire that blocked it, and so to upgrade from 32 to 64K, you go in and you clip this wire.

Booch: Oh my, so the memory was there already.

Humphrey: It was there already. I mean it basically would have been much more expensive to do it any other way, so they had done that, and this is the way all features were done. They had no connection. They would price the feature, and you'd price the parts independently. And so the whole structure, and the way IBM was making money was all of a sudden exposed by these guys, and what they were doing was, they were buying machines of their own, purchased machines. They'd pick them up, and they'd de-feature them. They'd take parts off of them, so people weren't buying parts from us, they were getting parts and pulling features and adding features and so they were charging much less than IBM did, but making a bundle of money.

And so these guys sued IBM, and the reason they sued us wasn't because of the way we were doing it. They sued us because, in moving to the 370 line, IBM was claiming that the maintenance documentation revealed proprietary information and we could not make it available. That was an out in our 1956 consent decree, that if-- that we had to make maintenance information available, but we were not required to make proprietary information available. And so they concluded that this was proprietary information and therefore we didn't have to make it available. And so the lawyers, when they heard this, they came back to me, and said, "Well what's the story, what do we do? If this is all proprietary it's got to be all right." And I said, "So that's the case. People tell you it's proprietary information and so you can- we can go ahead and defend the lawsuit." And I said, "Oh yeah, sure." I said, "Well let me just check." So I called the hardware staff, the engineering staff, in Corporate. I was in Corporate.

And I said, "I understand this stuff is proprietary. If you've got something that's proprietary, that's maintenance information, you could show me, so I can see what it looks like." So they brought me the maintenance documentation for one of our new disk files. And I said, "Show me what's proprietary." So we started to go through it. And the maintenance material had nothing in it that revealed any real technical design data or anything that was in any way proprietary. And so this was their best example of proprietary documentation. So I sent a wire- a note out immediately to all the division Presidents, and I said, "Have your people go immediately through all of your maintenance documentation and tell me immediately about anything that is proprietary, so we can, you know, examine it." And so all of the

division Presidents did, except one. The one was Bob Evans. And Bob answered me immediately, and said, "Of course it's proprietary, we're not going to bother."

So I basically ignored that and went with all the others. And no one found anything that was proprietary. It was extraordinary. And so I told the lawyers, I said, "You're not going to be able to defend that lawsuit," and went through it with them, and they agreed. So we had to basically change the policy on how we were going to handle maintenance material on the 370, and make it available and we did. And so, but basically the issue was an enormous problem, and so that was one of the ones.

Another one came up-- There were a bunch of lawsuit issues. There are a couple more I want to mention. One came up, they were coming up with a new display. I think we had the 2250, and they were coming up with the 3270. I've forgotten the numbers, but it was some new display that was going to be on the 370. It was only coming out with the 370 line. They were coming up with new equipment. And so when we came up with new equipment, we could use new policies, new prices and everything else, so the development people had a lot of flexibility.

And so they came up with this new display, and if you remember, I talked about plug compatible displays. People were coming out with plug compatible memories and displays and all kinds of stuff, and we were getting our clock cleaned in the display business, so other people were selling displays for the equivalent of two to three months of the IBM lease price, and our displays were not selling at all. The market almost moved completely away from us. And so the new display actually was priced, and when they came out with the announcement, they were dropping the programming support for the old display. So the new ones, you could obviously do whatever you want, but they basically were saying that in the release of the operating system that was going to support only the new displays and nothing else, so people had to move to the new release.

The engineers said, "We will no longer support the old display." And I said, "Why is that?" And the answer was, now this is what the lawyers told me, they were reviewing it with me -- I was involved in a lot of this stuff. And so the lawyers were reviewing it with me, and they said, "Well it turns out, what they're saying is, we don't have to spend any extra money to support the old hardware." And I said, "That's certainly true." And so if that's the case, well it's completely defensible. And they agreed, "Yes, that's true." So okay, the question is, are they spending extra money, would they have to spend any extra money to do it? And I said, "I'll bet you what's happened is that actually the old support was there already, and they had to spend extra money to take it out." I said, "If they did that, would that be defensible?" And they said, "No, it would not." And I said, "Well you'd better go back and make sure, because my bet is that they spent money to take the support for the old displays out, not the other way around." They went back and discovered that what I had said was true, and they would have lost a multi-billion dollar lawsuit over that.

Booch: Ouch.

Humphrey: Well that's the kind of stuff we were facing all the time. And so this was the problem with people being so convinced that, if this is the way it should be, that it's the way that it is, even if it ain't true. We had to look at the facts, and that was really kind of amazing, but it was certainly the case. Another one was an interesting one. It came up-- I would get calls from these-- I had meetings with my policy development worldwide folks. We would come together for meetings.

I remember we had meetings out at Montauk Point out there, and we'd typically have it in the late summer for about two or three days. And people kind of objected to what we were spending. I said, "Look, it's the only way I can stay in touch with all these guys worldwide," but I made it clear to them, and these were guys, we'd have come from all over, Europe and Asia and everywhere else. I made it clear to them that if you have a legal problem, call me. Any time, day or night, I can get it fixed within-- usually within an hour, and I can call the Chairman, and I did on occasion.

So, like, this is what we've got to do. We can make decisions instantly and get it done. And it was marvelous, so we had a marvelous pipeline, and these people would, they'd pick up the phone and call me. And I got a call on, it was again on a Friday I think, late Friday. We got these calls at terrible hours, so you had the weekend to do stuff. So I got this call, and I think it was like, on a Friday afternoon, late in the US. It was from the Netherlands. And it turns out this guy was saying, we've got a problem with a customer, where there's a person where they added a memory in the machine from a competitor.

It's an IBM machine, and our people, our maintenance people were unwilling to work on the machine, because it's a competitor's memory, and we've got to do this and that, and they had all kinds of problems with it. And it turned out that the EC lawyers were arriving Monday morning to examine the situation and see what was going on. And the machine wasn't working. And so he said, "What do we do?" I told him right then, I said, "Get maintenance people out there over the weekend and get it fixed."

Booch: As I recall, the phone company, which had a monopoly at that time, had the rule that you couldn't attach any non-certified equipment from them, and there was this great lawsuit, I think around the same time over somebody that just wanted to put a little device on the handset to make it more comfortable to listen to, and that broke open the notion of having third party things, so what would the year of this have been?

Humphrey: This must have been about '72, '73, '74, along in there.

Booch: Very nice.

The IBM PC Story

Humphrey: It was in the '70s. Another thing came up along in here. I'll make sure I'm getting everything in order. Yeah, I remember there was great push, and it must have been in '77, '78, '76, right along in there anyway, that the personal computer was beginning to rear its head, and you know, a lot of this little homemade stuff, and things popping up, and there was a great pressure that IBM had to have a personal computer. And Frank Cary, who was, at that point, CEO. This was before Opel came in. Frank Cary was still CEO, so it must have been earlier than '78, but it was in there somewhere. In any event, Frank Cary said, "We're going to develop a PC. So he pushed the divisions and he got no response. And everybody was sort of looking at it, but they didn't do anything. So he set up a brand new group whose job was just to develop the personal computer.

And he broke them out and he said, "They are reporting to me." And basically they reported directly up through to Frank Cary, a direct pipe. They were in one of the labs, but they had a direct pipe to Frank. And he said, "You can do anything you want. I want a PC on the market in a year." And so that's what they did. And they got the hardware going, they were moving at a great speed, they did a great job, but

they didn't have any programming. And so they were trying to get-- they needed about 20 people to start putting together a basic operating system and get something going. If you remember, they had broken up programming. If I had had 4,000 people working for me, I could have put 20 people on the PC in a minute.

But the programmers were all distributed to all the different hardware groups. And there wasn't any operating system group or anybody that would pay any attention to these guys, so they couldn't get a nickel. And it was like what I ran into with the TSS thing. I don't know if I mentioned this, but when I was doing the TSS, doing that programming, I couldn't get any programmers out of IBM at all. We got the guys out of the Advance Systems Development Division, and then we hired a whole lot of guys from CDC and CSC to do it. No, it was CSC and another one of the big programming contractors. So we had contractor developers developing the TSS. So in any event, so this was the case here with the PC, and so they literally had to go talk to somebody to acquire the software. And basically it was all done, there were a lot of us objecting to this, but Frank basically said, "No, we're going to break down, we're going to break any rules to do it the way we want." And so they didn't put any constraints on ownership on the chips or on the programming.

Booch: Was there a lot of push-back from that notion?

Humphrey: Oh yeah, a few of us objected, but we knew that Frank was behind it so no one objected very loudly. They were giving away the Crown Jewels, but we couldn't get an ear. Frank was not in the least bit interested in listening to whatever we were saying, so that's just the way it worked. So it went out and basically the business went with it. Another thing. I'm getting nearer the end here, but I've still got a ways to go.

Booch: Sure, I'm up for continuing here as long as you wish, because to get to-- we can take a break for lunch if you want, and come back to it, because I've cleared my afternoon now.

Humphrey: Oh you did?

Booch: Yeah, so I'm good to go. Would you like to take a break for lunch now? This is maybe a good stopping place.

International Business

Humphrey: Okay, yeah, I've got one more story on the corporate staff job, and then I moved off that. Let me tell you the last story on the corporate staff job, and then we'll come back in 1979.

Booch: Yep, and we can finish up the-- You have the Iran and China thing around that time as well, too.

Humphrey: Let me hit that right now.

Booch: Okay.

Humphrey: As part of my job as Director of Policy Development, I was also on a council they had on dealing with overseas sales. I mean, marketing to all kinds of countries. And we had lots of problems. I mean, how do you deal with people in the Middle East, for instance, it was a real tough problem, because a lot of these countries, there's the *baksheesh* [bribe], as you know, you've got to pay stuff. And IBM absolutely refused to do that, and so in a lot of the countries, though, that was absolutely required. You literally couldn't do business without it. We typically worked through agents, and so we didn't do it. We basically had agents that would handle it. It was sort of transparent, because they were doing it all under the table. Iran was a tough one to deal with. We actually had an IBM business in Iran.

Booch: The Shah was still in power then, is that correct?

Humphrey: He was in power and you had to deal with his cohorts. The business was just terrible. It was extremely hard. We weren't making any money. The guys were going through in a meeting with Frank Cary, this was in the boardroom, and I was there. The European marketing people were presenting the story about the problems we were having in Iran and what the guys were doing to us. They had these special rules and they were doing all kinds of stuff to us. So Frank kind of slammed his fist on the table and said, "Dammit, let's just get out." He said, "It's become a place where we don't want to do business." So we did, we got out.

We literally got out of Iran. It was before the Shah was dumped, before any of that stuff. We got out in time. It was amazing. So that was something. About that same time, the Chinese, of course there was no real interaction, but Nixon had been to China and so things were sort of opening up a little, but we really had no business in China at all. It was done through Hong Kong. One of the things that no one had really cottoned onto was that IBM's prices were different in different countries. So we were basically pricing to the market, not the cost. There were all kinds of different tariffs and all kinds of shipping costs. So the whole thing about how things work internationally was extraordinarily complicated. The Chinese appeared from behind the bamboo curtain. They wanted to place some enormous orders, big disk files and computers and all kinds of stuff, millions of dollars of stuff for the Bank of China.

Booch: Who was the premier of China at that time?

Humphrey: I don't remember.

Booch: I'll have to dig that one up.

Humphrey: It was probably Mao. Mao was still in, I think. I think it was before the thousand flowers bloom and all of that stuff. In any event, so the Chinese appeared and I couldn't believe it. We got the call from the AFSEA (Africa, South East Asia, I think) people. What the order was they had given us was just extraordinary. When you really sat down and went through it, they had taken and they knew the prices in all of these different countries and they were buying different products in different countries and shipping them and doing stuff. They optimized their costs at our expense.

It was just unbelievable what they did. We couldn't do anything about it, because it was all standard stuff. They could order it and they could do what they wanted. But I was just amazed by what they had done. They had sort of figured out how to get around essentially all of this stuff that IBM had, our strange pricing to make a bundle of money. They figured it out and they had just taken it. They cleaned our clock. It was

just amazing. Also about the same time, we were dumped out of India. We had been in India and we had our own business there and they basically concluded that we couldn't be there and be wholly owned. We had to be majority owned by Indians. IBM just basically got out of India as well.

Booch: So this was pressure from the Indian government itself?

Humphrey: Oh yes. They had their own rules. They would not allow you to be wholly owned. We were running into that in all kinds of places. So I got involved in all of those things. We had a lot of stuff going on. So that's probably a good place to break right now.

Booch: Let's do that. Let me stop the recording here.

<break>

Day 2 PM, June 18, 2009

Booch: We're back again and we'll pick up where we were, but I believe, Watts, you said you had another anecdote regarding Jay Forrester you wanted to cover.

Another Boston Story

Humphrey: Right. When I was in Boston, for some reason they talked me into running for chairman of the IEEE computer society. I'd been an IEE member and an IRE member since I finished college and they were merged, so I was an early member.

Booch: If I can clarify, what year would that have been?

Humphrey: That would have been in about 1956-1957, along in there. This is actually in 1959, a little bit later, they had me run for president of the Boston computer society. I wasn't terribly interested, but I decided to do it. I didn't think I had a prayer of winning. I didn't know anybody. No one knew me. But I was elected, unfortunately. I think my name, Watts, may have gotten a lot of people lined up. But in any event, I had to leave. But through some connection, whether it was from that or not, someone got hold of me and said, "Hey Watts, you ought to meet Jay Forrester." Jay was the guy that invented the core memory. He was there. They used the core memory at Whirlwind Computer.

Barbara Humphrey: Dudley Buck was the inventor of core memory. His picture was on--

Humphrey: No, no. Dudley Buck was cryogenics. That was very cold computing or low temperature computing. But anyway, so I went over and had lunch with Jay Forrester. I frankly don't remember much about the lunch. He was a very pleasant gentleman. But just a minor point that I happened to meet him was kind of interesting. I've been exposed to lots of people.

Booch: Remarkable. By the way, speaking of the Whirlwind, have you been out to the museum in the West Coast and seen what pieces of the Whirlwind they have?

Humphrey: I did. I did stop off there. I mentioned something we had, which unfortunately we didn't keep. When Barbara was there at the Whirlwind, they were just converting from the old-- they had CRT tube memories. The old memory on the Whirlwind, the original memory stored dots on I guess it was charges on the Williams tubes and they scanned them. They had to keep rescanning them to keep the memory. So they had this enormous bay of Williams tubes that were storing memory. Of course, enormously expensive. So they were replacing the Williams tubes with the core memory. When they pulled out the Williams tubes, for some reason, they cut some of them up and they were cleaned out so they didn't have any of that display stuff on them. So they had these glass jars. Barbara got one of them and we used it as a cookie jar for quite a while. It was about eight inches in diameter and about 14 inches tall. I wish we'd kept it. So that was that one.

The Program Pricing Flap

Another thing I don't think I mentioned was the ACM meeting. I missed that. It was on my list I'd written in. This was in about 19-- had to have been 1967. It was before we priced programming. Everybody thought that programming pricing was going to come. There was lots going on in the subject. I was invited to sit on a panel discussion it at an ACM meeting in Edinburgh, Scotland. There were a whole bunch of questions and there were a lot of things that they wanted me to talk about. So I figured that I probably ought to be prepared to talk about so I actually prepared an answer for what would happen if we priced IBM's programs. Everybody was saying that if we priced the programs, the hardware cost would be cut 40 percent and all this kind of stuff. That was nonsense. So I put together an answer to the question and wondered what to do about it, how to get it approved, because I wanted to say something but I didn't think I ought to just go out and say it. I knew enough to know I shouldn't. So I called Hillary Faw. Remember, I mentioned his name?

Booch: Right.

Humphrey: Hillary Faw was the pricing guy on the FAA and later. He was then the Director of Policy Development on corporate, a job I later had. So I called Hillary and I told him what I was going to do, that I had put together this statement and could I have him look at it and was it okay? So I did. In fact, I sent it to him. He looked at it and he called me back and said, "Yes, that's fine. You go right ahead." So when I got over to the meeting where the questions came up about pricing programming. I said, "Well, there's a real misunderstanding here. People are saying that-- I won't tell you whether we're going to price it or not, because that's a policy decision the IBM company will make, but I should tell you that there's a lot of misinformation on the subject.

People think that if IBM priced its programs, there would be an enormous reduction in hardware costs. The estimates that ranged in the press were in the 40 to 50 percent range. If you want to figure out really what it costs and what might happen to the price based on cost, and I can't tell you, of course, how we'd price it, but based on cost, there's no way you'd get a cut of more than maybe two or three percent in the hardware costs. The reason is because the way you price stuff covers costs. If you look at the costs for IBM and take out the marketing people and all of that, but you look at the general cost, what percentage

of the IBM population are programmers? It's less than three percent. So 40 percent is crazy. There's no way you'd do that.

"The costs aren't going to come down any more than about three percent." This hit the press. When I got home, I had all these calls from IBM management. "What got you to you say that?" I told them it had been approved. I reviewed it with Hillary Faw and they were really upset that I hadn't reviewed it with my boss, George Kennard, straight nonsense. I didn't say it to anybody, but I think people realized it. George is a prince of a guy, but he wouldn't have said, "No, don't say it." He'd have had no problem with it. But in any event, I did get by without a wrist slap coming. But I thought that was kind of interesting, because people had a wholly unrealistic view of what might happen. In fact, when we did price programming, the hardware price was actually reduced by 3%.

Booch: By the way, you mentioned that the programmer population was four or so percent. What was the entire IBM population around that time? Do you have a number?

Humphrey: I'm afraid I don't remember. I know when I retired in 1986, it was 420,000 people.

Booch: I do recall the IBM archivists have that information, so we can dig up the numbers. But I was just curious as to four percent of what. I'll see if I can find that number.

The FS System

Humphrey: Yes. It wasn't a big number. The FS system, let me talk about that. In 1970, after Bob Evans became division president, a fellow named George Raden, a real bright guy, had been in programming. He'd gone into research. I think it was George. I know he was involved in it, but I'm not sure if he was the guy that actually invented it or not. But it was basically a new way to architect machines. The new architecture was basically designed around the programming system and designed to minimize and simplify and accelerate programming. He figured how to do this and minimize it and make very simple machines that really worked like lightning. It was a marvelous idea.

He demonstrated it, showed how effective it was, and people got all interested in building machines on that architecture. They were proposing it to Bob, this was right after he got to running the division, that that should be the foundation for our future line of machines. So he asked me to lead a small group to look at the feasibility of doing this, the hardware architecture feasibility. I was a hardware architect too as well as the software guy. So I did. We went to Hursley and we talked to a whole bunch of folks. We went back and met with Bob and said, "Technically, there's no reason you couldn't do this in terms of building a system or building a family of systems." It looked like a marvelous way to build compatible systems and a range of stuff.

So Bob launched-- this was when he decided they did the-- this is, I think, before we did the-- this was after we were doing the 370, which was a modest patch-up on the 360. We were already doing that. Our future system on down the road--five, six, eight years down--would be the FS system. It would be a brand new system. We would no longer stick with the 360 line. Whether that was a good idea or not, he didn't ask me. The question was: Could we build one? I wasn't real sure it was a good idea, but there we go. In any event, it had a big crowd of people. Dick Case was brought in to lead a big task group to sort of lay

out the FS plan and they laid out a strategy for it and how they were going to do it and the way it would be programmed.

But essentially, an emulating system. But it was to be a colossal system. However, it was not designed for small systems, those guys really suffered. That's why they later had to have-- they broke off a small systems group and there'd been no preparation done for small machines. So basically the preparation for IBM's future was all tied to FS. So IBM basically, for a period of about two to three years, development was essentially out of gas in terms of preparing for the future. They were working on FS. That's where all the money went.

Booch: Give me the years again?

Humphrey: 1970, 1971, 1972.

Booch: This was just right around the time that DEC was releasing their PDP-11. How much was that on your radar?

Humphrey: Oh, it was killing us. That's what the small systems guys were worried about.

Booch: Got it.

Humphrey: I believe the FS was aimed at that. It wasn't aimed below it. The DEC machines weren't really low end. They were small, but they weren't the System 3 kind of stuff. So the really small business stuff got essentially killed by this. So FS got started, and I wasn't much involved then. I just had to run the Endicott lab. So I wasn't involved in the FS stuff, but I was asking questions, as were others. At one point, they got it to the point where they were able to making some estimates of performance. What they discovered was performance on COBOL would be slower than the 360 by quite a bit. Nobody reacted to that except Marketing blew up. So they escalated up. After about two to three years of a lot of work, they killed FS. But I thought it was an interesting thing, because it was a whole new system, hardware and software. I think it might have been a disaster if they'd done it. It didn't have the big-- did we lose you? Are you still there?

Booch: Yes, still here.

Humphrey: We basically lost an awful lot of time in preparing for a competitive world, and I think that badly hurt IBM for the next 10 or 15 years.

Booch: So what I'm hearing you say is that this was a foreshadowing of its stumbling in the mainframe market. Was this the beginnings of that?

Humphrey: Yes, absolutely. I think that is true.

Booch: Can I go back to a question? You mentioned something about COBOL. It occurred to me we hadn't talked about when that hit on the scene. Did you ever have any contact with the CODASYL committee or Grace Murray Hopper and those folks, as COBOL was evolving?

The SHARE Meeting

Humphrey: Actually, I never met Grace Hopper in those days. I met her later at SEI. But I had a lot of involvement with the SHARE and the GUIDE committees. I remember multiple meetings with them. As a matter of fact, the first talk I made to the-- was it the SHARE committee? It was a big meeting in Miami very shortly after I took over for programming. It was a meeting of a couple thousand people. It was an enormous place.

Booch: SHARE used to be a really big meeting.

Humphrey: Yes, I think it was the SHARE meeting. It was a great big crowd. There must have been 2,000 people there. So I was the keynote speaker. The thing they pumped me up to talk about was the PL/1 language. I did talk about PL/1. It basically almost caused a riot. The COBOL people were up in arms. The Fortran people were up in arms that we were going to kill their languages. I basically went through what we were going to do. We weren't going to kill any of those languages. We were moving to PL/1, which never took off, as you know. It was a nice language, but either it wasn't far enough or it went too far.

The MFT System

So it didn't permit [an] easy move from Fortran or COBOL. I basically approached the programming guys, the PL/1 guys to put together a manual so they'd have a Fortran users' PL/1 manual and a COBOL users' PL/1 manual that would actually write in a better language. But they never did. One other thing that happened was the MVS system. Was it MVS? No. MFT. This was a multiple programming system that was our biggest version of the 360. The original OS/360 came in three versions. It was a single string program or it could multi-program in two ways.

Booch: Do you think that was the MVS, the multiple virtual system?

Humphrey: No, that's not what it was. It was a multiple one with fixed partitions or something. It was MF or something like that.

Booch: The OS they called OS-360 MFT. What that stood for, I don't know, but I'll look it up.

Humphrey: It was basically for programs with fixed memory partitions and MVS was with variable memory partitions. MFT was a very fixed way to run a fixed number of jobs that were rather limited in size.

Booch: Here we go. Multiple programming with a fixed number of tasks. MFT. That's what it stood for.

Humphrey: MFT. We originally announced it, but when we later re-announced, we dropped it. When I went through with all the re-announcement and everything, because the MFT wasn't terribly attractive versus MVS. So I got into battles about that. The MFT, the marketing people were up in arms. We had to have MFT and they were going to give us added money for it. We had to do it, etc. So I agreed and we took the money. I had the guys put together a plan.

I had to start with the requirements, exactly what we wanted to do. They couldn't agree on the requirements. The developers had a limited amount of time and money. It had to be ready in November, or it wasn't going to beat the window before MVS, which was the following June. So they had to get that done. They were having these big arguments and they came to me, "How do we do this?" I said, "Well, that's fine. You either agree with marketing on the requirements by the end of the month or we're not going to do it." I gave them a deadline.

So they agreed by the end of the month. Everybody compromised. They built the system. It was, in fact, ready in November and nobody bought it. While they agreed on the requirements, it turned out that nobody wanted what they had agreed on. So building what they wanted, I'm sure would have been a lot closer to MVS. But we wasted a bunch of money and time on that, but we did it. So I learned from that that you've got to be awfully careful when you lower the boom on requirements. And that is not a good move. You could meet the schedule but miss the market and lose a pot load of money.

The MVS Review

The MVS system, which was delivered on schedule in I think it was June 1967, we were coming out with it in the spring. It was like March-April for early testing. They were really nailing it down. I had been pushing them almost, because of the MFT experience, to really get people in from the field, experienced competent people, and I wanted to have some reviews to go through and to tell us what they wanted. We could not afford to screw it up. The development guys said, "Great. Sure. We'll do that."

The marketing people said they were happy to do it, but we had to send senior system engineers. They weren't willing to have customers come. So we were going to get senior systems engineers to come in. Everybody agreed, but the development guys said, "We're not ready yet." So every couple of weeks I'd call them and say, "Are you ready yet?" They'd say, "No, we're not ready." Then one week they said, "It's too late." I said, "You lose." So we had it anyway. I remember the meeting. We were in Poughkeepsie. All these managers were there to describe this thing. They had a couple of architects and other fellows.

There must have been a dozen really top flight systems engineers out of major high end accounts. They were listening to this. Every so often they'd ask a question. With few exceptions, the managers couldn't answer the questions. They'd have to call the programmers. I remember one particular question somebody asked. It concerned the restart mechanism. I've got to come back and tell you about the check point restart in a minute. But it concerned a restart mechanism. It was restarting the printer. It was a very limited restart. It was a printer restart. The problem was that a very frequent mistake was a JCL error. I don't know if Fred [Brooks] mentioned it when you interviewed him, but he concluded that JCL was the worst programming language ever invented and he invented it.

Booch: He, in fact, admitted to that mistake. He went on to say that the reason that he believed it failed is that they only accidentally developed it and never really treated it as a full language. As a result, there was no intentionality for it at all. So yes, Fred took responsibility for that one.

Humphrey: It was extremely hard to program [in JCL]. People made lots of errors. A very common error was in printing. People would write the JCL in such a way that it would skip a page every line of print, instead of a line every line of print, which wasted a lot of paper and they had to kill the printing. The printer restart was supposed to solve that problem. The systems engineers said, "What happens under these conditions now that you've got the restart?" They had a way to do it now that they were using – a work-around. So they wanted to know what would the whole MVS system do under these conditions? It turned out the MVS system, once they did the printer restart, MVS would--

<disconnected>

Humphrey: Did I lose you?

Booch: It just dropped. The last words you said were, "When you did the printer reset" and then it clicked off.

Humphrey: Okay. When they did the traditional printer restart, the MVS system would basically run right through it and start printing all over again, skipping a page every line of print. Everybody concluded that. There were about three or four or five things in the MVS that the system engineers pointed out that would be totally unacceptable. If the system would have been shipped, it would have been a disaster. So the MVS guys agreed, "Yes, we've got to fix all of them," and they did. They fixed them all and they got it out in time.

But the real lesson of the story isn't that. The real lesson of the story is that they never did that again, calling in an expert team like that to review a system or a product before its release. They never did it. I pushed them to do it. I would have had to lower the boom and make that a big issue, to set up a whole procedure, which I did not do. I probably should have. But I was amazed. The laboratory view was that the marketing stuff was an annoyance. "We'll build it. We'll build our system and we'll do it right." I was just astounded. Anyway, that was that one. We really didn't have any reasonable kind of requirements review system at all. Another thing I was going to mention, we had lots of problems with the quality assurance community. This is again in the 1966-67 timeframe. Maybe it was 1967. Yes, I think it was 1967. We were putting out like about nine releases in the first year in 1966 and something like that, a similar rate in 1967.

Release 15/16

The quality assurance guys nonconcurred with every release. They'd come in. I'd have to go to meet with the group executive to overcome their nonconcurrence. And we did. They'd come up with a list of defects that you had to fix and there were these 50 problems here and all of that. So we'd agree to go through the list with them, identify which ones were truly critical and get them fixed. So we went through that every time until we got to release 15/16. And in 15/16 was the new version of COBOL. We were replacing the old COBOL with a new up-to-date COBOL. So as we put out that, for the first time the quality assurance people concurred with the release. It looked great. So we shipped it out and it was an absolute disaster. <phone ringing> Just a minute.

Booch: Sure. A junk fax again?

Humphrey: It turned out the new COBOL was a complete replacement for the old COBOL and they had not gone through the release to test the new version against all the defect reports and test cases for what they called APARS, which is the programming fixes for the prior version of COBOL. They had not gone through and corrected it and tested it to make sure it would work on all of them. So the new COBOL had all the problems that had been defects in the old one.

Booch: Were these primarily problems in the language or problems in the compiler that IBM offered?

Humphrey: It wasn't language problems. They were in the compiler. They were essentially defects in the system and it was kind of amazing that they would get repeated, but as I discovered by looking at defects, people tend to make similar errors all the time. And programmers do, in particular. So pretty much identical errors show up all the time. But in any event, they had not done that. And as a consequence, whether you know it or not, I don't know, but when people install new programs, the rate of installation varies enormously. For instance, operating systems go in very slowly and communication systems do as well and file systems are very slow. But new language compilers typically just snap in. So they get put in instantly. So thousands of people had just installed this new COBOL right away and it blew up on them. So we had an absolute disaster. In any event, we basically shut down our Time-Life laboratory to go fix that one. It was interesting; the whole issue of determining the quality of programs, the typical testing didn't do it. You really had to do something well beyond that. That was a severe problem. Now let me see where I am now.

Booch: You had some issues on compatibility and the story about Gartner that you wanted to cover as well.

The Compatibility Letter

Humphrey: Yes, I was very concerned about compatibility. This was in 1970 when I was in Endicott. So I put together a letter to Spike Beitzel, and I think I mentioned it before. It turned out the letter or excerpts from the letter were picked up in a book. The book was called *Big Blue: IBM's Use and Abuse of Power*. The author is Richard Thomas Delamarter, published by Dodd Mead in 1986. Appendix 1 is called *A Voice Crying in the Wilderness*. They say right at the top where he says, "The most important problem at IBM that its customers face today is how to enable its diverse and incompatible product line to communicate easily." Remember, this is 1986 now that the book was published.

Booch: And if I recall, if I'm not mistaken, looking at this, wasn't he on the government side on the anti-trust case? So he had a little bit of an axe to grind, did he not?

Humphrey: I don't know who he was. I don't. That's very likely true. In any event, he said, "One who saw it all coming in 1970," that's an overstatement. I'll talk about that a little bit more, "and vainly tried to stop this proliferation of incompatibility was IBM's W.S. Humphrey of the Systems Development Division, Endicott, New York laboratory. He addressed the question of." and then I've got three paragraphs that he quoted. Shall I read them in?

Booch: Sure, please. That will be great.

Humphrey: “Whether a compatible product line should be our data processing group objective.” That’s the one that I’m addressing. “This is such a fundamental issue that I have taken some time to prepare an answer.” Now remember, this letter was from me in 1970 to our IBM’s group executive running all of our development groups. “Compatibility is a major and growing requirement of our customers.” This is in 1970. “Compatibility is most essential for the advanced and highest potential applications environment. I strongly recommend that the DP group adopt as its strategic goal the achievement of an operationally compatible product line. Each of our systems and product strategies should be measured against this goal. Our planning and testing should be so directed, and our organization and management systems should be established toward this end.” I did elsewhere define operationally compatible. I wasn’t just talking about using data or compatible programs. I was talking about an entire compatible operation where the whole installation was able to move stuff around and that sort of thing. I go on.

IBM’s concern was that even then, IBM’s product line had too many layers of incompatible systems. And it goes on. “In any case, where growth is blocked by an incompatible barrier we find that the customers are either slow to move or reassess their IBM decision versus competition. In the first case, the growth of 360 Model 20 customers into the DOS environment has been almost completely stopped by the incompatible nature of these two systems. Similarly, the customer growth out of the 360 Model 40 marketplace. It is clear that smooth growth—“ Oh, I see. This is talking about the growth out of the 360 marketplace, that’s from DOS to MVS. “It is clear that smooth growth is an absolute requirement that is growth within a product line and growth between generations of a product line. In each case, our customer is concerned with the ability to install new systems preferably without extensive conversion.” He goes on to say, “Humphrey could see what was coming, that these systems would soon be hooked together into large networks.”

So the next paragraph says, “I believe that interconnected systems, as well as interconnected networks of systems will be of growing importance in the 1970s and a major factor in the 1980s. This being the case, we should recognize this requirement for 370 and the system 3 Extension. It is an absolute requirement for FS,”--I’ve just talked about FS--“which would be introduced in the late ‘70s and would be operational in the marketplace through much of the 1980s. Operational compatibility in that timeframe will be required across the complete product line. For growth, for instance, it will be completely unacceptable to introduce a system like the 360 Model 20.

Similarly, the introduction of a System 3 with an incompatible architecture would preclude any significant upward growth, would be recognized as limited to the small single installation customer but unacceptable to a large customer with many small operations. By the end of the 1970s and into the 1980s, we must offer a completely compatible line from the terminal and device to the smallest and the very largest of our processors.” So that was what I thought and the letter didn’t even get answered, which I thought was interesting. So some of us could see what direction the market would be moving in the future but IBM management didn’t pay any attention. In any event, so that was that. Now another thing I didn’t mention, when I was on corporate staff, remember I had the job or I was looking sort of at the IBM Business System?

Booch: Right.

Vision of the Future

Humphrey: One of the things I was asked to look at was sort of where we were going as a business. We had various task forces looking at stuff. I remember we had a task force looking at terminals and communications and stuff. I ran a host of these things in terms of the future. I remember somebody saying, “How do we define an intelligent terminal?” Remember, this was in a meeting in about 1972 or something like that in corporate. I said, “The definition of intelligent terminal, I believe, is one that will run MVS.” People laughed. They thought that was crazy, but it’s true. But what was interesting to me was--

Booch: You foreshadowed the notions of workstations.

Humphrey: Yes, exactly. But I could see sort of the networks coming. I could see the whole idea of vastly increased power and remote workstations. I could see the enormous needs for compatibility and dynamic networks and all that kind of stuff and the tremendous growth of programming. In fact, I was arguing strenuously when I was talking about the cost, customer’s cost. Where did the customer’s money go? Customer’s money basically was going increasingly to software. I showed these charts of where their money was going in terms of spending on support systems and application systems. They weren’t spending it on hardware; they were spending it on writing programs and installing operating systems. They weren’t spending it on buying programs, because they weren’t priced. We just started to price them. There wasn’t much money in it yet. But the cost, more than 50 percent of their cost was now in stuff other than hardware and it was going up exponentially.

So my contention was we were looking at the wrong target. We had to move. No reaction, no interest whatever. But even though I could see that, I couldn’t put together enough of a picture to make a case. I must admit, the future surprised me as much as everybody else. I wasn’t the only one. I had a few allies, but not a whole lot, way back in those days. So anyway, then-- let’s see where I am. So vision of the future. That was that.

Technology Assessment

Okay, now I had basically, I was on corporate staff and I was beginning-- I had been there now, this was in 1979. It was getting kind of late. I talked to my boss and I had been working with the lawyers. I was involved in all the lawsuits and I was advising the lawyers and all this stuff. But I wasn’t building anything. I wasn’t learning anything. I wasn’t doing anything. I had a modest staff and a big, prestigious job, but it wasn’t really terribly interesting anymore. I’d sort of been there long enough. I learned a hell of a lot. So I was talking about where to go next, and I was saying I would like to go back and run a lab or do something like that. I knew I wasn’t going to be king anymore. So I got a call to see if I would come down and work for a guy, Art Anderson, who had just taken over as the new group vice president running all the development divisions, development and manufacturing. So Anderson actually came out of research. He was kind of a funny guy to go into the job. I don’t think he really did it very well, but he was a very sharp guy.

So I went and talked to Art. He wanted me to be director of technology, basically technology assessment. I guess that’s what it was, Director of Technology Assessment. He told me about how they’d done assessments in the various research labs. Basically, it was a way to get groups to assess themselves of where they stood versus technology versus what was going on in the world. Were they really doing superior work? Where could they improve? It was a way to get groups to sort of wake them up and think about how they could move and what they could do better. So I agreed to do it.

I joined him and I had to build a small staff of folks. I got people out of various labs to come and work with me temporarily on a one year assignment. I got people from Europe as well. I got a European at one point. I got a Japanese. I would get them out of the country managers. I'd talk to the country managers and see who they had, they recommended. It was an upcoming person. I got a guy out of Italy once. I got one out of the Netherlands. I got one out of Japan. I had people out of various labs in the U.S. So I had a bunch of folks.

Booch: If I can ask, where was this office physically set up? Where you working at this time now?

Humphrey: It was in White Plains, New York.

Booch: Okay.

Humphrey: It was the headquarters of all the development folks.

Booch: Got it.

Semiconductor Quality

Humphrey: One of the things that I hit almost immediately turned out to be very interesting. I learned a lot about it, because I learned what assessments were and how powerful they could be, self-assessments. We learned, by looking at some of this stuff, that the costs of our semiconductor chips, as manufactured in our Burlington, Vermont plant, it cost us more to build them than it did to buy imported chips from Japan. That was just not competitive at all. So Art and I went up to Burlington and met with the management team. He took a very straightforward position. That was either you can get competitive in manufacturing chips or we're going to close Burlington. Burlington, Vermont must have had 30,000 employees.

It was an enormous place, made all of our chips, but they weren't competitive. They did a self-assessment. I went back and forth helping them pull it together and learned a hell of a lot about chips and chip manufacturing and quality. They actually brought the yield of chip manufacturing up to a very high number. I don't know what it was. I certainly don't remember. I probably knew at the time, but their yield had been like 20 percent. I think it was up to like about 80 percent. It was purely quality management. That's all they did. But they really went through it. They crawled through it all. They looked at what everybody else was doing. Of course, the actual yield achieved was a highly classified number. Nobody would tell it. It was competitively extraordinarily important. But they actually got to where they were the lowest cost manufacturer in the world. They did it fast. It was amazing to me how quickly they were able to absolutely change their stripes. They became the leader in memory chips for some time.

I learned a hell of a lot about quality and quality management and the W.E. Deming stuff and how you manage quality. That served me in good stead later with the software community. As a matter of fact, one question came up and that was: Could we take any of the lessons from that and use them in software? Art asked the question. As I say, Art had some very interesting perceptions. So we pulled together a group of experts from software and the semiconductor group and had a session on it to see was there

anything we could gain from that, between the software and semiconductor guys. The software reaction was, “No, there’s nothing we can learn from this.” There was no real interest in looking any further.

Another thing, I mentioned Gartner, Gideon Gartner. While I was working for Art Anderson, increasingly we were having people leave to go work for Gideon Gartner, who just started this little group. They were basically a group spying on IBM, trying to get clues as to what IBM was doing. A good friend of mine, Tom Crotty had joined Gideon Gartner. IBM was very upset with Gartner and they were trying to figure out, “How can we shut these guys down? They’re doing something illegal, etc, etc.” So Art asked me to go meet with Gideon, which I did. I had quite a discussion with Gideon on this whole thing. Interesting guy.

As you know, the Gartner Group is doing a lot of stuff these days. But I’m sure they’re still doing much the same. They were sort of a spy shop, checking on what everybody was doing. But nonetheless, I met with him, but I didn’t really make a whole lot of headway. We ended up suing him anyway. I don’t know quite what happened. But in any event, that was sort of a quickie. Next, Art Anderson-- I think that takes me through all of that at this point. Yes, next Art Anderson was moved on. I don’t know where he went. Jack Kuehler replaced him. Remember I mentioned that Bob Evans had said there were two guys who were on the list to be IBM president/CEO down the road?

Booch: Right.

Humphrey: It was me and Jack Kuehler. Well, Jack had been-- when I was running the Endicott lab, he was running the Raleigh lab. A great guy. We worked together a lot, because we did a lot of printer interaction work and so we spent a lot of time working together with our groups. We had to develop a printer for one of their terminals. The people couldn’t agree. So basically, Jack and I said, “Okay, we’ll put them all in a room.” So we shut them all in a room in Endicott. It must have been a half dozen of his guys and a half dozen of mine. We basically said, “We’re locking the door and throwing away the key until you agree on a strategy.” So we went in and met with them periodically. But we gave them really firm marching orders. Every so often they’d come out with problems and questions and we’d keep going.

Booch: Did you burn some little things so that it either made a white smoke or a black smoke when you were done?

Humphrey: That’s right, pretty much. We finally got-- the idea of just locking people up and giving them a clear charter and you’ve got to go back and help them. Then frequently they needed some motivation and some facilitation. But they did a great job. Did I tell you the PC story?

Booch: No. You did a little bit. We hadn’t quite gotten to the announcement of the PC. Around Art, you also were going to mention Nick Donofrio and John Akers and where they came in.

Software Quality and Process

Humphrey: Oh, yes. Well, Nick Donofrio actually turned out to be, he was just an administrative assistant or sort of a technical assistant to Art. He was just sort of in the office. He was just a young guy. I knew him, but I wasn’t involved with him at all. But John Akers was one of the-- I think John was head of the-- I don’t remember what he was. He had been head of the marketing division, I believe. In any event, he was

there reporting to Art Anderson at the time. I think he'd come on to Art's staff, sort of assistant general manager or something or assistant group executive. So I knew John quite well. He and I had a lot of interaction. Very crisp guy, but he was a marketing guy. In any event, that sort of is that.

Art Anderson moved out and Jack Kuehler now came in as group executive running all of development. So Jack called me in and said, "Watts, what are you doing here?" I was on this technical assessment job, Director of Technology Assessment. I said, "Frankly, with Art gone, I'm not sure." He went through a career talk. "What do you want to do?" At this time, this was-- what would I say? It would have been about 1982. So I was about 55 years old and Jack was the same age. I was off the promotion track. He basically was asking, "What do you want to do for the remainder of your career?" I said I really didn't know. He said, "Well, let's think about that." He said, "Give it some thought," so I did.

Booch: Did IBM have a mandatory retirement age at that time for its executives?

Humphrey: No. Senior execs had to retire at 60, like the CEO. I don't know if the senior group executives did or not. But the CEO, Tom Watson had basically decreed that. A lot of people thought that he put it in place so that Learson would be forced out in two years. I think Tom was scared that Learson would be a little roughshod and he was. Tom had real insight. So I think it was just the CEO had to retire at age 60. Everybody else was 65. In any event, I had five to ten years to go. I didn't know what I wanted to do. So Jack said, "Well, look, there are a couple of things. You could either run one of these programming shops we've got that are running the lab computer shops." He said, "That's one. Let me think about some others."

A few days later he came back and said, "One job I'd really like you to consider is to take over as Director of Software Quality and Process." It's the group in Poughkeepsie. We were living in Darien, Connecticut at the time. It would have been impossible to commute. But I looked at it and thought about it and I said, "You know, maybe that isn't a bad idea." So I agreed to do it and I did. We didn't move to Poughkeepsie. Barbara never wanted to move to Poughkeepsie. So we moved back to Chappaqua and I had about an hour commute each way to get to work in the morning. I had about 60-70 people, 70-some people, included a bunch of people working on process work, quality work, etc, some very good folks.

CI 105

It was a fascinating staff job, but it got me back into programming and it got me back thinking about programming quality. There were a bunch of things involved with that. One was fascinating. It was called CI-105, Corporate Instruction 105. IBM had put out a corporate instruction on quality. It was for the company's products, all products. It required that each new product have better quality than its predecessor IBM products and better quality than the best competitors, the leading competitive products.

Booch: Did you have a sense for how to measure quality then, so you knew what you were going against?

Humphrey: Good question. The hardware guys had a measure that everybody had been using. It was called RAs or repair actions. The maintenance community that did the hardware maintenance, the maintenance people had lots of measures of RA. The thing that was absolutely astounding about actually measuring and tracking quality and managing it step by step, they had to project what the quality was

going to be before they could announce the product. They had to show what they were doing. People had to review and concur with it. They tracked it. When people missed their quality goal, because they had goals over time, when they missed the goals, they had to come in with action plans and so it was really managed and people were really paying attention from the top. On one file system they had a quality improvement in RAs of over 1,000 times.

The numbers were absolutely extraordinary. They had another thing with the shipment of IBM's largest machines out of the factory. For instance, out at Kingston they'd ship a large computer system, it would fill a 42-foot moving van. It was a lot of stuff and that's how they moved it nationwide in big moving vans. Worldwide, they typically went by air. It cost too much to let it float around. But when they'd arrive early on, the machines, they'd go to put them in and they'd be missing stuff and things wouldn't fit and all kinds of stuff. So they started putting in tracking on the installation quality on these systems. It took us, typically at that time it would take a week to 10 days to install a new machine.

Amdahl at that time was installing machines in about three to five days. It was an enormous problem. When you have a great big system and a machine room, people can't shut their whole business down for 10 days. Most of them don't have backup systems, at least they didn't then. They don't have all the machine rooms, so they really had to be able to pull them out and put them in quickly. So they set a goal, and the people in Kingston thought it was crazy. The goal was we're going to have defect-free installations. They were to track every defect on every installation and come back and figure out how to find it or prevent it. What was amazing to me, before I retired, which was in 1986, this would have been in the early 1980s. Before I retired in 1986, about 80 percent of their large system installations were going in defect-free. They would arrive at the customer's office on late Friday and they would be up and running Monday morning. It was an extraordinary achievement. And they did it with a very routine way. They just basically measured every defect. They went back to find out what caused it or how they could find it earlier and they laid in procedures and practices to put that in place. They basically went through with that and that's what they did. And boy, did it work. That was a marvelous lesson for the software community. So CI-105, basically that was what they were after.

Measuring Software Quality

What had happened before I got the job, just before I'd gotten the job, one of the division presidents had been pumped up by his software guys to give a story to corporate management, which was the president, CEO and others, about how CI-105 could not apply to software. And so he'd gone in and made this story. And Kuehler and I'm trying to remember who was the president or the CEO then. I think it was Opel was the CEO at the time. He made this presentation, and Opel and Kuehler basically looked at him and said, "No, you're wrong. It applies." And the division president was replaced within 30 days. That got a lot of attention, and that's when I got the job. And so a big part of my job at the very beginning was putting in place CI-105 for software. So I had some very good guys, and Bill Florak was working for me at the time.

They had a quality report, which we got, which had maintenance data, but they had a whole bunch of other things too. So of course, the report they were putting out monthly that they were shipping to the labs. We had a lot of data, but I didn't know what the report was for. So the guys came to me one day with this report for me to sign, because I had to sign it every month. And I'd read through it, and I didn't know how anybody in the lab would use it or what it would be for. So I asked. They said, "Oh, everybody needs it. They've got to have it." So I said, "Oh, okay." So when I went out and did a lot of lab reviews on quality and what they were doing, I asked about this report and what they did with it. And so every month when

they got the report, they put together a staff group to go through it and figure out what was wrong with it and argue with it. That's all they did with it.

It was just basically a source of debate, and so they basically would come back and argue with the staff about how they found errors in it and stuff. So when I got back and they brought me the next report, I said, “I want to make one change.” They said, “What's that?” And I said, “I want you to put right here, 'This report will now be quarterly.'” They said, “Oh, you can't do that. There'll be all kinds of problems.” I said, “Well, do it, and let's see what happens.” So they did. No flap at all. No one objected particularly, but they still were going through the recycle. So after two quarters, when they brought it in for me to sign, I said, “I want to make one change.” They said, “What's that?” I said, “This is the last report.” So we shut her down, and it didn't cause a ripple.

But in any event, that was sort of a side light on this. So software quality was being addressed in a rather superficial bunch of ways. So CI-105, we had to figure out how we could really measure quality of software. So we called together people from a bunch of the labs, put together a working group, technical guys, led by Bill Florak and his team in Poughkeepsie. And they worked for a week or two to figure out the right way to measure software quality. They had a debate with really top flight people from all the laboratories, and they could not agree. So you'd think it was a dead loss, but I had a very good crew. So when the meetings were over and they basically didn't have anything, they said, “We think we know how to do it.” I said, “Okay, let's put together a proposal, a proposed CI-105 quality measure.” So they did. And we put together a pretty good one. I don't know if you want me telling you what it was?

Booch: Please.

Improving Software Quality

Humphrey: Okay. Well, basically, it was focused on the defects, the number of defects found in the first six months of installation of a product, and the number found during the first five years, product Y. The number found in the first six months had to be fewer than the previous version. We had all kinds of problems in terms of installation rates and everything, and so people had to make plans based on history. They had to project out what they expected in terms of numbers, and then measure against their plans and all that sort of stuff. But they put together a series of measures. It was a cumulative defect report, is what it was. Everybody objected to it. I sent it out to the labs. A number of them were kind of reluctant, so we talked to them and bent a lot of arms. I basically got most of the people to buy it, except the Poughkeepsie lab. The director of the software lab or manager of the software lab in Poughkeepsie would not buy it. I said, “What do you want to do?” Basically, when I went out, I issued it to them, and I said, “Either come back and agree to do it, or come back with your proposal for what we should do.”

And so basically, most of them reluctantly said nothing, or they said, “Okay, we'll do it.” But Poughkeepsie came back and said, “No. We're not going to do it and we don't think it should be done.” I said, “Okay.” I called Jack Kuehler, who was then president of the IBM company. I said, “Jack, I'd like to have a meeting with you next Thursday.” He said, “On what?” so I told him. He said, “You bet. Bring them on.” So I called the lab director and I said, “We've got a meeting with Jack Kuehler next Thursday.” He said, “Oh, what on?” I said, “On CI-105, and you're going to tell him why you don’t think we should measure software quality.” He gulped and bought it. So we got the measures in place. We never had to go to Jack at all. I called him back and said, “Thanks.”

We put it in place, and we had it in place for two or three years before I retired. We cut the number of defects like about 50 percent a year on the measured programs. It was just extraordinary. People learned an enormous amount. They learned the rates at which people installed. They learned the rates at which errors happened. I don't know how they did it. Basically, it was all intuitive. No one had a whole lot of data. We did learn one thing. I can remember one product, and we did learn a whole lot. It was something that I think Ron-- no, it was Mike Fagan came up with it, I believe. I think it was Mike Fagan came up with the idea. And Ron Radice used it. And that was the-- I don't remember if it was Mike Fagan or Ron Radice, but in any event-- it was Ron. When Ron had worked as a program manager. That's right. And he later worked on my staff.

He looked at some data and figured out that most of the defects clustered. The defects in big systems cluster in a limited number of defective modules. And if you really want to focus on improvement, the thing to do is to identify those modules by gathering data as you're testing it, and then thoroughly inspect the defective modules and clean them up, following Fagan's inspection methods. And Ron used it on some products in Kingston that had been enormously successful. So that was one of the things. And so at one point, my quality guys-- I, by the way, in the quality job, quality and process job, I had studiously stayed out of the announcement cycle. My predecessor, who I think was Al Pietrasanta as a matter of fact, he went down to run the SEI, IBM’s Software Engineering Institute. We had an institute for systems engineering, basically an educational facility in New York City, so he went to run that. But Ron Radice had worked for him when Al had my job.

So they'd actually been reviewing every product announcement, and there was an enormous amount of busy work, just looking at every announcement and getting involved in announcement reviews. I concluded that there was no particular advantage to us. There were lots of guys involved, so I stayed out of it. I basically said, “We're not going to be involved. Take me off the approval list. You just go through quality assurance and the other guys can do it. You don't need my approval.” So they did. And on one product, it actually had been subcontracted out and then brought in for testing. I think it was an MVS product. Our guys came in and said, “This is a dog. You literally can't let it go. It has nine defects per KLOC (thousand lines of code). It's going to be a zoo. It'll be a real failure in the marketplace. So you've really got to stop it.”

So we went and got hold of the product manager and talked to him. I told him, “Really, what you needed to do,” and I told him the technique, which actually identified the defective modules in this thing, and get it fixed. He said, “No, we don't have time. We've got approval. We're going to go.”

So I said, “Okay. I don't agree.” So I put a note to that effect, copied the quality assurance people, the service people, the marketing people, everybody else. “We believe that your product is defective and should not be shipped at this point.” And so all the concurrences that this guy had previously had were withdrawn. All of a sudden, nobody agreed that he should ship the product and he had to come to me. I said, “Here's what you do,” and I had him go back and follow the Radice method. I said, “I bet you'll bring it way down. I suspect on that product you'll find at least 300 defects.” He said, “Oh, no, no, no. Not a chance.” So they did it and they got past 400.

It was amazing. So they finally got it, they were all done. The product was really pretty damn good, and I finally agreed. Then they played bloody hell getting all their approvals from everybody else back. It took them months to get the service people and the quality assurance people and the marketing people all back in the boat. Just telling them Watts agreed wasn't enough. And so thereafter, any time I sneezed

about quality, everybody paid attention, because they know if I disagreed, everybody would call off their approvals, and it wasn't just a matter of getting me in agreement any more.

The Amdahl Story

So it was easier to work with me, because I could tell them exactly what to do. So as it turned out, I got enormous power out of that, which I had not expected, but it was really quite effective. Another thing that happened next. We had the CI-105 stuff in place, and it really worked extremely well. Matter of fact, one of the reasons I learned that it worked so well. I later went out to review a lab we had in San Francisco, I think it was. It's a group that IBM had bought. IBM had started going into all kinds of miscellaneous stuff, looking for ways to grow. They had a satellite system they were going to put in place, and they had the Prodigy system.

That reminds me, let me tell you about Prodigy. I have that story. I really didn't get involved much in the satellite thing, although it was a real hot button of Bob Evans, but it really was never a particularly good business. But they did actually get into that business. But they'd also bought this company that was in telephone switchboards. This was a little company, out in San Francisco, and they made these programmable machines, and they had a lot of software in them. So I'd been asked to go out and look at it. They were having quality problems with this system. What they had, they had about 1,600 modules of code, a pretty good-sized programming system, about half a million lines of code.

Booch: In what language was this?

Humphrey: I don't remember. It was very likely C or something like that. I don't know. It wasn't IBM traditional stuff. But they shipped the product, like two or three years before, and their maintenance costs, the defects they were getting, were just out of sight. They were spending a bundle on it. So the head of the operation asked me out there to review it and give him some advice.

And so we did, we went over it. And the first thing I said was, “What data have you got?” They said, “We don't have any. What kind of data do you mean?” I said, “I want to know the defect data, error data, anything like that.” “Oh, we don't have any of that.” Well, it turned out they did. As some engineers had gathered some of this data, they had data on the modules and how big they were. So we had them put together the data. It was sort of a flank speed effort to get this stuff together and look at the 1,600 modules. What we found was that 3 percent of the modules had over 50 percent of the defects. I think it was-- yeah, 3 percent had 50 percent.

I believe something like 86 percent of the modules had had no defects in three years. So 14 percent of the modules had all the defects, and 3 percent had half of them. So I said, “What you want to do.” and so they did. They decided to go do it. That's what they did. It was amazing. It really did clean up their costs dramatically. I didn't stay long enough to find out exactly what happened, but I'm sure it made a big difference. While I was there, however, one of the ladies, who was one of the engineers there, I don't know, she was a manager or something, was Naomi Trapnell. She turned out to be Fritz Trapnell's wife. Remember Fritz who used to run OS/360 for me?

Booch: Yes.

Humphrey: He had left IBM from the Hursley lab, and later joined Amdahl Corporation as their director of programming. He was now the director of programming for Amdahl. Naomi invited me to come over and have dinner with them. I said, “That's great.” So I did. We weren't going to talk business at all. There was nothing anti-competitive about it. So I went over and we had a lovely dinner, and Fritz at one point said, “What in the world did you do to quality?” “What do you mean?” He said, “Starting about a year ago, our customers started to tell us that the quality of IBM's programs have gotten dramatically better.” That was CI-105. My reaction was, “When the competitors start to praise the quality of your products, you're doing the right stuff.” It really made a hell of a difference. So I had a nice chat with him.

Booch: Did you tell him much about what was going on?

Humphrey: No.

Booch: Did you reveal-- I was wondering.

Humphrey: No, I didn't tell him. I just told him that we had put together-- I told him the objectives, the quality had to be better than its predecessor, but not any of the mechanics or the measurements, or anything like that. That was our secret, so I couldn't tell him that. I think I'm down to the last item on my list, with Erich Bloch.

My Outrageous Commitment

Booch: Yep, which then brings you to the end of your IBM-ish time. I do want to hear about your vision for the future bit, and then we'll transition to the SEI.

Humphrey: Great. Okay. Well, at that point, I was now 59 years old and I was debating what do I do next? I'd been talking to Jack Kuehler and others about where I go. Jack was president. I had a lot of friends, and was wondering what to do. Erich Bloch was an old friend of mine, who had left. He ran the Fishkill lab, the semiconductor labs. He'd been a VP of various stuff. Great guy. And he'd gone down to be the head of the National Science Foundation. I guess that's what he was. Reports to the U.S. President. He'd gone down, that was his job. So I knew Erich, I knew a lot of these people and I was debating what to do. I remember one day, one of our daughters, our number three daughter, really urged Barbara and I to go to a seminar. Did you ever hear of Werner Erhard?

Booch: Can't say that I have.

Humphrey: The EST system, have you ever heard of it?

Booch: I have not.

Humphrey: Well, he had this self-help kind of thing where you analyze things and people think about it. It's for people who are kind of troubled and misfits and that sort of thing. He had a seminar, all kinds of stuff he'd do where people would go there and they'd have big groups of people and have all these

discussions. He was kind of a spellbinding speaker, lots of interesting stuff about how people ought to work together and work out your problems and resolve them all, this sort of thing. It was kind of sad, because he wrote these books and had this great story, and was really very convincing, but it turned out later he had all the problems he was talking about in his own personal life. He had terrible problems with his kids that he'd never worked out. In any event, a lot of this stuff he pushed was quite persuasive and quite interesting.

So we did go to a seminar he had on commitment. I think it was just a day, maybe two days, I don't know. It was in New York City, and our daughter went with us, and we went down to this thing. Basically, the commitments they were pushing were people committing to do more of this Werner Erhard stuff. But they talked through commitments, and they basically were talking about, “You ought to make an outrageous commitment, something that will really capture you.”

One of the ones they were talking about was stamp out world hunger and that sort of thing. None of that was terribly interesting to me, but basically, as I was thinking about it, I thought, “You know, what I really need to do is to figure out what I am going to do for the future and think about that.” That's what I was struggling with. And I'd been struggling with what I wanted to do next. Did I want to be a consultant, did I want to be a this or a that? I concluded, “No, let me think about this. I want to commit to do something.” I had been concerned, because I'd spent a lot of my time out reviewing different-- I'd meet with customers, I'd look at our labs, and the software business had not really progressed very damn well. We'd made a lot of progress in IBM, but the rest of the world was nowhere. And they weren't getting anywhere. They were just sort of coasting along, writing programs the same way for 40 years, and it didn't look like it would ever get better. It wasn't 40 years then, but in any event, the programming methods and the practices, people were doing crappy work.

There's no other field where you bang out code and fix it and test. Automobiles, when you manufacture them and fix them in test, they're lemons and they always will be. You'll never fix them. They've got all kinds of problems with them. You can't make semiconductors by fixing them in test, and the software community wasn't focusing on building quality products before test. They were counting on testing, and as you and I know, a test is a very inefficient way to find defects. It takes forever, it's unpredictable and it doesn't find very many. So I felt the programming world was in terrible shape, not only in terms of quality management, in terms of schedules and estimates and planning and controls and tracking the whole business. It wasn't run like a business.

And so I decided what I was going to do, I was going to make an outrageous commitment and I was going to fix programming or change programming and that was what I was going to do. So that seemed like a great idea. The next question is, “How the hell do I do that?” I talked to a bunch of people about it and I talked to Jack Kuehler about it, and Jack said, “Well, why don't you go talk to Erich Bloch?” I said, “That's a great idea.” So I got hold of Erich and went down to Washington and I met with him, had quite a discussion. He said, “What you ought to do is go to the Software Engineering Institute. They're just starting it up.” He said, “I know the guys there. I could contact them and you could go talk to them, see if you'd like to do that.” I said, “That sounds like a great idea. Could you write to CMU for me or to the SEI?” He said, “Sure.” He said, “You write the letter and I'll sign it.” So I put together a letter and sent it down to him and he sent it. And I then got a call, went to visit with the SEI, and the rest is history, so there we are. So that's how I ended up at the SEI with my what I'd call it, my outrageous commitment to change the world of software.

Booch: So tell me the state of the SEI at that time. Larry Druffel was of course in the midst of this, and I haven't heard his name in it. Where did he come in on the scene?

Humphrey: Okay, well let me back up.

Booch: Sure.

The IBM Prodigy System

Humphrey: I realize there's one thing I forgot to mention. I didn't mentioned Prodigy. The Prodigy system, one of the guys, an old friend of ours, Ted Papes had been a systems executive. IBM had started-- you know what the Prodigy was? Ever hear of it?

Booch: I don't remember, no.

Humphrey: IBM started this online system--

Booch: Oh, that Prodigy. Yes, I remember now. Tell me more.

Humphrey: An online system, and it allowed people to come in and use all these fancy features that would allow them to access encyclopedias and play games and do reference work and all kinds of stuff. It also allowed them to send messages, and Ted Papes was running it. The problem they had was, with Prodigy, that nobody wanted to use their encyclopedias and all that other stuff. They all wanted to send messages. So they kept screwing around with the system and the pricing, to discourage all this message stuff, so they'd start to use it right. I thought it was hilarious. The marketing people, and the whole crowd, basically, here the customers were telling them what they wanted to do. IBM had a lock on the whole message business from the word go. Nobody was paying attention to it. It was just extraordinary. So that's what happened. But anyway, so that was the Prodigy story. Okay, well back to your question on the SEI and the status of it at that time. Could we take a brief break for a minute?

Booch: Absolutely. Let me stop the recording. <break in recording> We're back again. Watts, you had some things you were going to say with regards to back to the beginnings of the SEI. I'll let you take it from there.

Joining SEI

Humphrey: Okay. Well, as I say, I came into the SEI, the way it happened, I wrote this letter for Erich Bloch and he sent it in. They called me right away. I don't remember the name of the guy who was director of the lab at the time. It was John Manley. They had a laboratory director that was not Larry Druffel. He'd been in place for, I guess, about a year. So the SEI had started in '85, and this is I guess in early -'86, so it hadn't been going very long. As it was going forward-- let me back off here.

First of all, the director of the lab was John Manley. He had me come out there, and I went out and visited. It was rather funny. One aside, [a] personal coincidence, I have a sister, actually, she's a step sister, who lives in Pittsburgh. I was debating whether I ought to call her or not and talk about my coming,

but I decided not to. I got hold of some friends we had that were right in town. She lived a little out of town, so I did not call her. I thought, I'll just go out and if I go out there, I'll get hold of her later. So I got down to the TWA terminal in JFK and was walking around waiting for my flight and someone called my name. “Hey, Watts!” There was my sister. She and her husband were coming back from Europe. “Where are you going?” Well, I was going to Pittsburgh and I hadn't told them. Boy, was I embarrassed. These things happen to you. If I'd known she was there and was looking for her, I never would have found her. But in any event, so I've had multiple events like that in my life.

So I did go out there and I talked to the SEI. They originally wanted me to be the director of the technology work. I didn't want to do that. I had what I wanted to do and running the technology group wasn't about to-- I mean, I talked to all the technology guys to see what they were doing, and they had all these great little things they were doing in terms of inventing little database approaches and this and that. None of them were going to change the world of software. I didn't see that they were going to help anybody. I didn't know why they were doing these projects and so I kept asking, “Why are you doing that? What's that going to do?”

The SEI mission was pretty clear. We were supposed to actually go out and fix the way software was done. I didn't see any of this technology work doing anything to solve software’s problems. And so the technologists had no interest in me, and I had no interest in them. So I went back at the end of the day to John and told him that I was interested in coming, but I didn't want that job. He said, “Well, we're going to have to figure out a job for you, because we certainly want you here and we need your help. Why don't you come out as a special assignment for me, and we'll figure out what your job is when you get here?” I said, “Fine, that's okay.”

Doing vs. Being

There's one thing I wanted to mention here by the way, and it was interesting, because I didn't really notice it until along right about in here. The transition. This was sort of the ambition point. The transition from wanting to be something to wanting to do something. It was an enormous change, because now all of a sudden, I wasn't worried about being king or being promoted to director or anything else. I had something I wanted to do. I had a commitment and I wanted to do it. It enormously freed my thinking. I was able to think rationally about what I wanted to do, and I realized-- it took me a while to realize it, but the problems I had with getting promoted to be company president, and the flap I had with all those people and that sort of thing, meant my career had topped out, it was probably the greatest thing that ever happened to me.

Because all the people that I had known that had moved on up— I was never going to be CEO. I might have made it to president, but every one of them that I know, they were great people and all of that, but they’ve all disappeared from sight. I had no idea what they were doing. I mean Learson showed up. He was working on the law of sea. He got involved in something like that and a few of the other people did, but not many and the Opels and the Careys and the Akers and all those guys kind of disappeared from view.

Here, you were at a point where you knew more than you ever knew before in your life and you have an opportunity to really do something. Thinking about *doing* instead of *being* really frees you up enormously. I realized a lot of the things that I had done, even at IBM; I had not— for instance, when I was running the lab, I had not really looked at how we could re-do our strategy and how we could really look at the printer

business in particular. I put Sy Tunis in as running the printer group shortly before I left the lab and he basically got competitive printers in and disassembled them and looked at how they were built.

He had a whole bunch of things - very creative, thinking about how to do this job better that had never occurred to me. I was kind of amazed. I didn't realize how much my passions and my thinking had been tailored or colored by ambition as opposed to really focusing on getting something done. And so, that's something I think about. It's hard for people to learn that, but it's enormously important. I was basically free. All of a sudden, I wasn't worried about being director of anything. I wanted to get some work done.

Booch: Would it be fair to say that you were entering, therefore, one of the most creative parts of your life?

Humphrey: Yes, I would say that's absolutely true. I've learned more in the last ten years, for instance, than I think I've learned in the rest. I mean it's amazing, except for the first ten years or so. But, it's been a marvelous experience and very rewarding. So, that's, I think, a very key point. In any event, so I went back. We agreed I was to join the SEI, agreed on a salary. IBM very generously made me an award on retirement, which was nice. So, everything looked great.

The SEI Director Was Fired

And so, I was still at IBM and all of a sudden, one day, I had some friends, dear friends who were at our wedding. My wife, Barbara, went to college with the lady, and they live in Pittsburgh, right downtown. So, I visited them when I was there with the SEI, the first visit. I got a letter from them. The letter had a clipping that John Manley had been fired from being the director of the SEI and that they were now looking for a director. Well, this is the guy I was supposed to have this special understanding with. All of a sudden, he was gone. So, my question was, "Do I have a job?" I had no idea. So anyway, the next day I got a telephone call from a fellow named Angel Jordan. I don't know if you know who Angel is.

Booch: I don't know.

Humphrey: He was provost at the university, [Carnegie] Mellon University. It's a big job, as you know; essentially running the university, but he called and said, "Watts, we know you were coming and you had agreed on a job with John, but we want you to come anyway." I thought that was just marvelous.

Booch: Well, that was good to know because you'd already quit your previous job.

Humphrey: Yes, and I had even gone to – I had had my retirement dinner and everything was ready to go. It was a relief. So, that's where it was. So, I arrived and then they say John Manley was out. The problem they had was that what the SEI was doing, the Air Force didn't want. They basically said, "This isn't doing anything for us." They reviewed it. They didn't like the programs we had. Nothing seemed to help them. They came out with a list of ten things that the SEI had to do. It was a directive. They fired the director. By the way, the technical people had also rebelled against him. So, he didn't have a whole lot of support there. He had been ousted. I think he's at University of Pittsburgh now. I was surprised. Remember I mentioned Jim Frame who worked for me at IBM?

Booch: Yes.

Humphrey: Well, Jim Frame had left IBM and gone to run programming at IT&T. John had worked for Jim Frame at IT&T. That was basically his programming background. So, I don't think he knew a whole lot about it, but he certainly was a good talker, but he didn't seem to have a vision for where the SEI was supposed to go. In any event, he was out. Larry Druffel was leading the search committee for the director of the institute. One of the questions to me was did I want to apply, and I said no and I thought that was kind of interesting, without a hesitation, but I didn't want to run a lab. I wanted to do what I was going to do, and that's what I said about being freed up. So, they were doing the search and I got involved in all of that.

The Software Acquisition Problem

Basically, temporarily, I was assigned to work for Bill Sweet, who was one of the top guys in the SEI at that point. Bill was a sweet guy and a nice guy out of industry. He didn't really have a whole lot of software background, but a very nice guy. He had a lot of contacts in industry and that sort of thing. One of the key things that the Air Force had really pushed us to do: they had ten items and the top one on their list was to come up with a way to help on the acquisition process for programming. They wanted us to work with Lincoln Laboratory, or rather with MITRE Corporation and the folks up at the systems division, that kind of systems division at the time up in Boston. So, that was their number one priority for us to do.

So, the question was, and I forgot who was acting director at the lab, I think— I think it was the Dutchman who came in. Boy, I've forgotten his name too. He was a wonderful guy, but he died prematurely. It's sad. He was the computer science chairman, but was the acting director. So in any event, I agreed. I'd take this study and Bill Sweet and I were going to do this study. We went up to meet with the folks up at MITRE and met with a whole bunch of people up there, with the systems division people and everything else, and they told us about a study, which I haven't been able to find. I never got it later. I didn't have a copy of it unfortunately. I'm sure it was classified.

But in any event, it was a study of 17 major software procurements from the systems division and they had all been in trouble. They had all been late. Average delay of I think 75%, which means a four year contract would have been delivered in seven years. They were all overrun by factors… So, everybody was in serious trouble. And so, we were supposed to look at that as sort of a test set as to how you do it better.

I said, "Well, how did you pick the 17?" They said the 17 were all our large systems. I said that means you don't have a selection problem. They said, "What do you mean?" I said, "Everybody's failing. So, there's no way you're going to be able to pick a good one out of this crowd." I said, "The problem is no one is doing it properly." I'll tell you; this looked just like what I wanted to do. And so, I kind of got their attention. I said, "Perhaps the thing we ought to do is to look for a way to identify what are the characteristics of people who are doing good software work?"

So, we agreed and so that's what we did. We sat down and we put together a whole list of how would the acquisition people identify vendors who were doing competent software work. So, we put together this list of 100 and some questions, and that's what we did. We basically had questions, and people wanted to

put in stuff that they were using - Ada. Oh, I didn't talk about the Ada design language at IBM. I don't know if I ought to put that in. Do you want to make a note of that? I can go back and do that later.

Booch: Sure. Absolutely.

Humphrey: So, the question that we looked at was we were looking for things that distinguished between good and bad performance. I said, "If you don't have any evidence that this practice is associated with good performance instead of poor performance, then we shouldn't just drop in good ideas here. That's not what we're after. We're after things that really will say these people are going to do competent work." And I said, "There's no evidence that I'm aware of that says using Ada instead of something else is going to give you good results." They kind of grumbled and backed down.

But, that was a pretty tough test because no one really had much. Basically, the only real evidence we had is what I have seen and what I did with the OS 360 because I knew what practices helped because we had put them in place and I knew what worked on quality because we had put it in place and we measured the difference. So fundamentally, with all of that stuff I had gone through over the years, I was able to apply it right there and it was marvelous. So, we put this whole thing together and we put together a list of close to 100 questions and then we got to an issue next, and I was really struggling with this.

I remember once I was sitting in the Atlanta airport. I was stuck coming back from somewhere. I had been on some trip and flights were canceled and I was delayed and I spent hours in Atlanta airport. The issue at the back of my mind, and we had all been debating it, was how do you actually take 60 or 80 or 100 answers to these questions and how do you rank people against it so you know who is better and who is worse. We didn't have any idea how to do it at all.

The IBM Assessments

Well, let me back up a step because when I had been running the Software Quality and Process Group, my last job at IBM, I had gotten very interested in quality. I had heard about Quality College. What's the name of the guy who wrote it? *Quality is Free.* I'm just looking for his book. I don't know if you can hear me there.

Booch: Yes, I hear you.

Humphrey: Oh, Lord. I don't see it. Oh, yes. Here it is - Phil Crosby. Phil Crosby had been a quality manager at IT&T and he had put together this book on *Quality is Free* and he had a maturity model, five levels, sort of went through how you were doing. I had actually gotten this and I went to this very shortly after I took my job in Poughkeepsie in the quality and process job and I think I mentioned to you that Art Anderson had talked about assessment.

Booch: Yes.

Humphrey: And so, Ron Radice and I had gone to the Quality College, Crosby's Quality College in Orlando - it's about a week - and the guy had gone through how do you evaluate an organization in terms of quality attitude. You don't look at specific quality practices; you look at attitudes. Did they believe in quality? Did they know what it was? Did they have a program in place? So, he had this kind of five levels.

It was essentially an intuitive set of levels based on how people thought about quality and what they were doing about quality without talking about specific practices. And so, Ron and I decided that we would do assessments of the software community using a five level framework, using Crosby's maturity framework.

And so, Ron Radice and Jack Harding, both worked for me; I had them put together an assessment program and they went out and put together the way we were going to do it. The first one we decided to do was in the San Jose lab. I remember going out there and meeting with the lab managers and the lab director, trying to talk them into doing this assessment. We were going to follow exactly the guidelines that Art had strongly recommended, and that was the assessment was not a report card that we were getting on them. It was a report card that they made to themselves. We didn't publish it. We'd use it for guidance and that was it. So, we did.

We went out there and talked to them and they really didn't want to have anything to do with it at the beginning. We spent all day with them. We talked to the lab director; it was fairly early, but he wouldn't do anything until we convinced his staff. Well, the staff, finally at the end of the day— We kept saying we weren't going to tell anybody. This was it. It was for you. We weren't going to report on you. So, they finally bought it. So, we actually did an assessment and it had a marvelous effect, the people really were motivated by it. They focused on it. We did a bunch of assessments on a whole lot of labs just as we did at the San Jose one. They were quite excited about it. So, we went on and we did one in Poughkeepsie and in Hursley, England and we did Kingston.

I remember Kingston we did fairly quickly, but it was kind of a shock because we went back and did Kingston again about a year later. They wanted to see how they had done. We went back and did the second assessment. The problem was when you do an assessment, you can't really assess everything. I mean there's no practical way you're going to do it. So, we had a team of people in there for a week or two. It was actually a couple of weeks. It was a lot of work. We report back to the management. Present the results.

In the Kingston results, what you do typically is you go and select certain random projects and you'd look at the selected projects to see what they're doing and what's going on. And so, we did the same. Of course, the next time we went back, we selected whatever their current projects were. It was a different number, and we got the results and they had gotten worse. Kingston had put an enormous effort into improving things and the assessment result came out that they were lower maturity than they had been before. It was an enormous downer. It was disaster. Fundamentally, it kind of killed the assessment program because really, it didn't work. It didn't connect to anything that they were doing at all and there was no way to really be objective about it. So, that was a bit of history I had learned.

The SEI 5-Level Maturity Model

So now, back at my SEI job, talking with the guys at MITRE, we now had a list of 100 things nearly that people ought to do if they were doing good work. I was sitting in the airport in Atlanta and I had this list. I was puzzling about it and all of a sudden it hit me - how about the five-level Crosby model? So, I decided what I would do would be to lay out the five levels and put the questions against them, and I'll be damned, it was no trouble at all.

Booch: A revelation.

Humphrey: So, a five-level model and the whole purpose of it was really to try to judge how are people doing. Are they really doing the right kind of stuff? We wanted to keep it kind of general. We didn't want to get into how they do any of this stuff. We didn't want the acquisition people telling people how to plan and how to do anything else like that. So, we wanted it to be fairly abstract. We didn't want to get down— It wasn't what I would call a measurement of the organization. It was an indicator of how an organization is doing and whether they were following the right kind of practice. It wasn't a model of the organization. It wasn't a measurement of it. It was basically a motivational framework. And so, we put that together. I put together a brief report on our first pass on this of the— I think the first model we had together; we got it together by late in the year, late in 1986 we had the first pass at this thing. About that time, Larry Druffel got a call. Larry at this point had become the director of the SEI. They finished the search. They really didn't like anybody who had applied and the search committee concluded Larry was probably the best choice. So, they picked him and he accepted.

Booch: And, he came from Rational. He had been working with us there for a while. The other small world story is Larry was actually one of my teachers at the Air Force Academy when I was a cadet.

Humphrey: Oh, is that right?

Booch: Yes, small world.

Humphrey: He was also at DARPA for a while I believe.

Booch: Yes, that's correct and I got involved with him because of the Ada Program Office. So, he went from the Ada work and then we hired him inside Rational and then he went on to the SEI.

Humphrey: Yes. Well, he had been in a lot of good places and did exactly the right kind of work. In any event, so we put this together, but Larry got a call from some general in Washington, in the Pentagon, who wanted the SEI to go evaluate what was going on at the Standard Systems Center in Montgomery, Alabama, the Air Force Base there - Gunter Air Force Base. And so, Larry came to me and asked, "Watts, would you do an evaluation of this operation for us?" I said, "Sure, but on one condition." He said, "What's that?" I said, "That I don't tell anybody." And he said, "What do you mean?" and I said, "Look, I go in there as an auditor who is going to report to the Pentagon. They won't tell me anything. I won't learn anything. I won't have any idea how they're doing." I said, "If you want them to do better, I'm going to have to go in and do a confidential evaluation. If they want to tell the Pentagon, that's their business. I'm not going to tell the Pentagon. I have to have agreement on that" and we have got to proceed on that basis.

I strongly urged that the general not beat them up but use the results of the assessment as a guide for improving. And so, he went back and talked to the general and the general bought it. So, I went to down to Gunter, talked to the colonel, told him what was up, and he told me; he said, "Look, we have all these studies." He said, "My closet is full of them. The last study we had made something like 50-60 recommendations. To put them in place would have cost $13 million. We didn't have that kind of money. There's no way in the world we could do it." He said, "How is this going to do any better?" I said, "I don't know, but we're glad to go through it and you can take a look at it. My guess is you'll come up with something where you can change what you're doing without adding a whole lot of money, but I don't know."

So, he bought it and we went down and did an assessment and it was great. We identified a lot of things that ought to get done. They put in place some action plans. It turned out that they put this in place and he actually went back and reported to the general what they were doing and what their action plans were and all of that sort of thing. There hadn't been a commander out of that lab that had been promoted in history. They've never been promoted. They'd always failed. This commander was a colonel and he was made a one star [General] and moved on. Unfortunately, that was a disaster because when he was gone, all of a sudden the new guy wasn't interested. He had no skin in it.

And so, they went through like six or seven commanders in the next five or six years and they never did anything on the action plan. They finally got a lady in who took a look at all this stuff and finally said, "Okay. Quit screwing around - do it." So, they actually did it later, but it took them a long time. But in any event, but the structure we had worked – what we put together at the SEI actually did work because now we could connect actions to our evaluations. So, when people got better or worse, we could tell them why. All of a sudden, we had something that allowed and motivated improvement and the CMM [Capability Maturity Model] took off from that. That's where it went.

Booch: Wow. It's interesting to hear the beginnings. I had never heard the beginnings.

Humphrey: Yes. It was very effective. It was quite amazing. Now again, I really want to emphasize, and people lose this. They just don't get it, and that is the CMM is not for the purpose of measuring an organization. It's not trying to characterize it in any detail in terms of a model of it. Refining it to get more and more precise is a mistake. You're not actually doing that at all. The whole idea was to motivate people to think about how they're working and how to improve it, and you want to keep it simple. It has gotten way away from that. I'll come back to that later.

So, that's what we did with the CMM. During some of the early CMM development, we put that together. I wrote my book *Managing the Software Process*. It sort of describes what we were doing. It was sort of the bible for it, at least for a while. I remember going out and giving a talk. Remember we put out this little technical report I mentioned? We put that together - an SEI technical report. This was like in 1987, I think, or early 1988. It sort of described the method and the levels and all that sort of thing. It was before we actually had published CMM we put this little booklet together - SEI Technical Report. I remember going out and giving a talk on it at AIAA - is that one of the associations?

Booch: The American Institute for Advancement of - what do they stand for? I'll have to look those up.

Humphrey: Aeronautics and Astronautics- something like that. But, I talked to a number of these conferences. I have it - Aeronautics and Astronautics.

Booch: Ah, that's the one, yes.

Humphrey: I think I talked to one of their meetings and I took out a couple of dozen copies of the report and I started my talk and I said, "Oh, by the way. I have some copies of the report here if anybody wants it." It was a stampede. All of a sudden, people rushed up to get the report and they were all gone in about three minutes because everybody wanted it. Of course, there were about 200 people in the meeting. So, it didn't work. But in any event, it was extremely popular. People were very interested in it and it took off. As you know, that's sort of what happened with the CMM and then the CMMI. So, that's what we did

there. As we got going with that, we had to put together a group to structure it, to put it together in a much more formal kind of a framework, which they did with the CMM.

The CMM Steering Committee

One of the problems I had was as we were going through it, the guys were putting together a model for what was the CMM. They were making it much more formal than the sort of maturity model framework that I had put together myself. That was fine. I wanted them to do that. And so, we called in a whole lot of people from the Defense Industry who were very interested in this. A lot of people were interested. So, they all came in to a big meeting we had at the SEI and our guys went through a presentation about what we were doing with the CMM, the initial version of the CMM. We damn near had a riot. There were a whole bunch of people who just objected violently to what we were doing. “You didn’t talk to us about it. How come, etc, etc.” So, we decided we would co-opt the people that were most vociferous and we did. We started inviting them. We put together a steering committee. We got the people who were really the most outspoken and most concerned and put them on the steering committee.

So, we brought them in to work with us on making sure it did really meet their needs. There were a whole bunch of meetings for that. I had participated in some of the initial meetings and the guys rather soon after— I realized that I couldn’t sit in on the meetings. It wasn’t that I wasn’t interested. I was dying to sit in on the meetings, but the problem was they’d go through something. I tend to think out loud, like a lot of us do and said, “Well, how about this” to bounce it off people. Well, the minute I would say, “How about this,” everybody would say, “Okay.” It was sort of I had spoken. It was engraved in stone. That was it. Watts has spoken. Let’s move on to the next. So, I discovered that if I sat in this meeting and said anything, all of a sudden, all conversation would stop. They’d take what I wrote, wrote it down as the bible and they’d move on to the next topic. And so, basically my presence essentially destroyed any participation and so I had to step out of the meetings.

So, I was no longer involved in designing the thing. I would review it later. I had lots of questions and stuff like that, and I’d poke at it. My own guys, I could work with them, but not with the whole community. A number of people in the group I was able to work with them fine. I had to be careful. In any event, so as we went through all of this stuff, I began to realize fairly early on as we started doing the CMM and the- the CMM in particular. It wasn’t the PSP yet. I was beginning to wonder because— I was concerned because this was really kind of abstract at a fairly high level. People were talking about how we could use this in a very small organization. No one knew how to answer that. We didn’t know. That particular question hit me. And so, I started wondering about that. How would you really apply the CMM to a very small organization?

The PSP Research

So I decided that what I would do would be to try applying the CMM principles to my personal work and so I did. I tried to use it on balancing checkbooks, putting together simple processes and that didn’t tell me anything. So I decided I’d better write some programs. I hadn’t programmed in years but I was still running the process program at the SEI and I decided I would start writing Pascal programs because I hadn’t written anything in a long time, so I decided to do Pascal. And so I started doing that and I was asking the guys at work for help and then getting some guidance because they were a lot more recent programmers than I was.

And so I-- that's how I started the PSP [Personal Software Process] and I just-- it was extraordinary. I wrote a few programs. I had just gotten started and Larry Druffel called me in one day and he said I-- he was thinking of making me a fellow and I said, "Well, what would that mean?" Well, the reason-- by the way, the reason it came up was that-- this was Barbara again. I had retired from IBM. My mother was living in Florida-- Sarasota, Florida with my stepfather and-- I think I told you my mother and father had been divorced. Well, my mother had married a gentleman, Allen Osborne, a marvelous gentleman, really a prince of a guy and they had come to live in Florida. It turned out Allen Osborne was an old friend of the family and he had been a pilot in World War I and he was my father's flight instructor in World War I. Taught him to be a fighter pilot. My dad must have been a wild man because they called him the German Ace; he cracked up five American planes.

So it was an amazing family connection. But in any event, they were living in Sarasota, Florida so we had-- when I retired we came down here. My mother died within a couple of months of-- within about eight months of when I retired and we came down for the funeral and decided we'd buy a house. So we bought a little house and we used it as a vacation home and so we bought that in '86 and this is about, oh, three years later, '89, I'm managing the SEI Software Process Program, my book had come out and it was, kind of, wintertime and snowy and not very pleasant in Pittsburgh and Barbara said, "I really would like to live in our house in Florida." It seemed to me that was fair but I didn't know how I could do that with what I was doing.

So I went to talk to Larry Druffel about it and he said, "Well, we could make you a Fellow." And I said, "What's that mean?" He said, "That means you can work on what you want to work on where you want to work." And I said, "And you'll pay me, too?" And he said, "Oh yeah," and I said, "That sounds pretty good to me," and so I agreed and he set up a whole Fellow structure and everything else and it went through a procedure and a committee and they made me a Fellow and so I got myself replaced. I got Bill Curtis to come take my process job and so I decided to come down here and work on the PSP. The CMM was going already. I wasn't really that deeply involved. I was concerned with it because it wasn't really nailed down and I wanted to make sure it didn't go off course, but they were doing their own thing. I got busy doing the PSP, so that's what I did.

Booch: Back to Bill Curtis, it's interesting that his name came up here because he had been involved in-- wasn't there some semiconductor consortium down in Texas around that time from where he came?

Humphrey: Yes, the MCC.

Booch: The MCC, yes, that's where he came from.

Humphrey: He was down there. He worked for Les Belotti.

Booch: Yes, Les as well. All these names come back. Right.

Humphrey: I ran into Les since and so he worked for Les Belotti down there in the MCC. I remember going to a meeting with him and visiting with him and they had quite an interesting little group down there. They were doing great stuff but they were having a real problem getting industry support, particularly for the software and that stuff essentially wasn't getting anywhere. And so Bill was quite interested, I think, in his job but-- so he did come and replace me. I think it was around in December of '89 I believe. I'm not

sure, in that timeframe. I think it was about January ’90 I finally quit that job and came down here and as I say, I wrote a total of something like 72 programs in Pascal and then I moved to Object Pascal and then I moved to C++.

They were all, kind of, statistical programs and, sort of, analysis programs, stuff that wasn’t particularly useful for anybody but it was stuff that I was interested in and measuring things and stuff like that. So, kind of, programs I made up with little requirements but I had a whole lot of practices I was trying to follow and I pushed reuse and as a matter of fact, I made reuse libraries when I got into Object Pascal and then C++, so I built these big class libraries and I had all this reusable code. I got to where I would-- I was writing programs where something, like, 80% of the code was reused. It was enormously effective and the thing that was most effective about it to me was several things.

I mean, I put in some practices for how I do it but the quality results were extraordinary because I was using the PSP. I was measuring quality. I was managing it and by and large my programs would have zero or one or two defects in compiling and testing. I’d have practically no defects and the programs ran. I mean, I basically never had problems. My reusable programs, I’d just turn them on and I’d plug them in and they would work and it would be great.

So I could build and compose these great big systems without any real trouble and I had a series of procedures for how you write reusable programs. And basically what I did, essentially I had a user’s manual as, sort of, the opening of the listing. I had, sort of, instructions on how do you use the program, what are the calls and returns, what are they and then there’s a series of warnings about things you can’t do and shouldn’t do but that was basically it and so these programs I discovered-- it was a dream. I was able to compose programs and stuff.

The whole principle behind the PSP because when I started doing it my very first programs-- this is when I was still back in Pittsburgh-- when I wrote the very first programs I decided that I was going to measure everything because I didn’t know what to measure, so I’ll measure everything. And so I decided to measure how much time it took me to do everything, how many defects I found and fixed at each step, how big everything was that I built and it turned out that’s all I had to do. I measured that and that’s all the measures I needed, that’s all I could get basically. That’s what you do. First thing I discovered the minute I wanted to measure though I had to have a defined process because you can’t define a measure without having a process. What’s a defect? You got to know where you-- when you found it and when is defined by your process.

So basically when I wrote the first program, I was going to sit down and write it and I realized I couldn’t do this until I had a process, so I decided to put together a process and as I got developing the process for my very first program, I realized I’d made my first mistake. So I backed up again. So I next developed a process for developing a process. So that’s what I did first. I put together a very simple process-- development process and then I used that to develop my software process and then I used that to develop my first program. If you remember, I told you that my-- the way I write programs-- the way I learned to write programs was when I was inventing instruction sets. So I would actually go back and change the instruction sets when I was writing programs early on.

Well, I couldn’t do that with Pascal but I could do it with a process. So I would write following the process and then I realized that when I was finished with the process that it-- I really didn’t like the way the process worked in these ways. I put in a little procedure for myself. It’s called a PIP, Process

Improvement Proposal, and so I, kind of, made a note as I was writing my programs of the things that were inconvenient about the process and so I tuned my process and I kept doing that. I had a problem because if I tuned it too much, of course, my measurement system was screwed up because I couldn't measure and track from program to program.

So I had a problem with compatibility there and that-- I had various versions I went through. So that's what I did and so I gathered data. I had data on every defect. I've got it all here. I've got all these programs. It was all done by hand. I didn't have any tools to support me. I just had the regular compiler. So I gathered all this data by hand which was a real pain but I found it didn't take very long. I was quite surprised how little time it takes to track your time and defects and measure size of a program. I mean, it couldn't have been more than 5% or 10% of my time, if that.

The Process Conferences

Booch: In terms of influences, I remember this was around the time of-- there was a lot of discussion about what's called process programming and Lee Osterweil and crew were talking about that and Barry Boehm was in the midst of it as well. Do you have any recollection of that being an influence or you influencing what was going on there?

Humphrey: I participated in those workshops-- the software process workshops and as I came-- I presented-- I participated in all of them. They invited me to come and I went there and I presented the maturity model at one of them and everybody got all excited about it and so I'd do a big talk on that. I used to go to ICSE (International Conference on Software Engineering) meetings and all that and I discovered surprisingly quickly that I was really off in left field. No one was where I was and I kept describing what I was doing and why and everybody was very interested in what Watts had to say but nobody ever picked up on it. And I remember we had a-- one of the software process workshops was in Yokohama in Japan. Isn't that north-- that the north island?

Booch: Yes.

Humphrey: I think it is, yeah. Went out there and a bunch of the guys were there. Of course, I'm the old guy. I was the old guy in all of this stuff, so I was out there at that and the Japanese were probably closer than anybody to where I was and they were, sort of, pretty much reflecting what I was saying. They're much more people-oriented. I was quite surprised. And-- but-- so I went to all of these. I remember one-- I think it was outside of Washington, after about five or six years of these and I was working in Sarasota at this time, doing all this work and Lee and-- by the way I didn't say I had put together a steering committee when I was running software process and quality at IBM and I had Lee Osterweil on it as a-- and Lee Osterweil and Bob Balzer were both on the committee. I had a bunch of academics that were coming and reviewing what we were doing and giving us advice. I had a whole tool group and everything else in Poughkeepsie and they were coming and they were looking at the Ada design language stuff. We were doing all of that sort of stuff. So I-- when I got-- I got working with them as well and we had a great time in this meeting and there was a meeting near Denver and I'm trying to remember the name of it. Brecken--

Booch: Breckenridge, yeah.

Humphrey: Yeah, and we were meeting in Breckenridge and that's where Lee Osterweil came out with his processes are programs and I came up with this CMM maturity model. Harlan Mills was involved then. I didn't talk about my interactions with Harlan at IBM but I did have some interactions with him there and so Harlan was all excited about it. I remember at the ICSE meeting where Lee Osterweil and I both presented our talks, Harlan got up and made some comment about, "This is the best session we've ever had." But he was a wonderful guy. I loved him. I used to meet with him. I'd go over to the east coast of Florida occasionally if he's here and go meet with him. I disagreed with his Cleanroom stuff, though. Not because of what he did but I disagreed with it because it didn't measure. He didn't gather data and measure stuff and he and I were not together on the quality principles of where we were going.

But in any event, yes, I had met with these groups and we were not on the same wavelength at all and the meetings, they'd go through all this discussion. They'd want to automate this stuff and they were trying to get tools to do this and that and a language for this and it was tech-y. It wasn't focusing on what people do and that bothered me because the way I thought about it was if I'm running a programming organization how was this going to help? And I didn't see that and I still don't. I mean, I think they're still going in that same loop. I don't know at all. I think it stayed mostly academic but I really don't know. I haven't gone back and looked. That may be unfair.

But in any event, one of the papers I submitted for this workshop-- we'd submit papers every year and they were reviewing them. And so I'd submit a paper every year and I'd get invited and I'd go and one year they sent out the reviews. I'd never gotten the reviews before but one year they sent out the reviews. This is for the one at Washington and the reviews all said, "Why do we want to hear this stuff?" No one was interested in it and so the reviews came back. They had no interest in it but I was still invited to go and it was-- obviously, I was going only because I was Watts Humphrey, not because what I would say was of interest to anybody and I decided this is not something I'm going to continue doing, so I stopped.

The PSP Course

However, one meeting I did go to, it was the one in Germany. There was one in Germany and there was also a meeting in-- where was it? In Berlin-- a previous week. Some kind of a process conference and Peter Feiler and I gave a paper on process terminology we put together. And when I was sitting in this process conference, it was boring as the devil and I wasn't interested. I had developed the PSP and couldn't get anybody to try it and I'd talked to some people. I think Mahdevi, a professor from McGill at the time, not sure if you know him but he was a professor at McGill. I'm not sure where he is now but I was chatting to him about it and he said, "You ought to teach a course." I said, "That's not a bad idea." And so he said, "Well, I teach a software engineering course. Well, why don't you come teach at McGill?" I said, "That's a thought." So I began to think about how I'd teach a course and the-- and concluded that it was-- that to do that I needed some kind of a text.

So I-- in the conference while all these people were talking, I was organizing my textbook and this was-- I forgot was the beginning of-- early in the year, like, February. And so going to the conference I had outlined my textbook and I spent the weekend in Berlin because I had to go to this other conference elsewhere. And I had wonderful time there going to look at the torn-down Berlin wall and I went to a concert every night and it was-- I had a great weekend. Unfortunately, my wife wasn't with me, so I was all alone but it was a great weekend.

But I had this book to design and I went to sit through the software process workshop the next week. I realized that I had no further interest in it. I was getting nothing out of these conferences in terms of what I was trying to do. So I didn't quit in any way but I had decided not to continue going, so I just stopped going to the process workshops or ICSEs. None of it seemed to have any effect. I'll get back to ICSE in a minute but I wrote the textbook. I got started in February. I had used PSP to make a plan for my textbook and I got a hold of my publisher who published *Managing the Software Process* and he was very interested in it. I put together a schedule. My book would take me till January to get a manuscript out and then I expected to submit it the following year and et cetera, et cetera.

And then I was deciding what to do to teach it and at the same time one of the guys that I had known at the SEI who was there as a CMU student in our masters of software engineering program was Howard-- Howie Dow and he had actually spent some time with me and was very interested in what I was doing on the PSP and when he heard I was writing a book on it and was putting together a course, he got a hold of me and wanted to teach the course and he was going to teach it at the University of Massachusetts. He worked for DEC and he wanted to use it at DEC and teach it there and he wanted to start in September. My plan said the book wouldn't be done until January but I had been using a PSP on writing and my PSP data showed me I would be through with the manuscript-- first draft of the manuscript in September. So I gulped and I told Howie that I'll get you the manuscript in September for your course.

Booch: It's known as eating your own dog food.

Humphrey: Exactly, exactly. So I basically said, "This is why-- you might as well prove it," so I did. And it turned out to be a bit of a push at the end because I ended up having to reorganize the book a couple of times. And I took chapter one and then made it chapter thirteen and I decided I needed a whole new appendix because I was using a lot of statistics. But I didn't want to write a statistics textbook. I had found statistics textbooks almost incomprehensible, so I put together an appendix A in my *A Discipline for Software Engineering* book and appendix A was, sort of, a you-drive-it statistics, how do you do this stuff. And the programming problems I put in the textbook-- there were 10 programming problems, eight of which were all just simple statistics, simple correlations and linear regression, multiple regression. I had chi-squared tests and approximations, a bunch of things.

So it was a lot of interesting stuff but I had to explain it to people so the people who knew nothing about statistics could read it and that's what appendix A in my book was. I learned a lot by writing it. Writing about statistics was fascinating. So in any event it ended being an 800 page book and I got the manuscript together from February to September. Now, that's pretty speedy and so Howie Dow taught the course. I had decided not to teach at McGill but to teach at CMU and so we put a PSP course together at CMU.

And so I started that in January and so I used my manuscript and I taught the PSP course at CMU. I decided to wait on submitting the manuscript. I was going to rewrite it after I taught the course because the only data I had before I taught the course was personal data. So I got Howie Dow's data. The first class he did worked well. He had lots of constructive comments. It was very helpful having somebody else do it but he taught the first course. There were some guys at Embry-Riddle University wanted to teach it, so they taught a course. I don't think in parallel with mine. They taught in the fall and others, so I had-- some guy at Howard University wanted to teach a course and he got in there and it was a disaster. He didn't really get it.

I guess about three or four courses were taught and I taught some courses at-- some companies got quite interested and they worked extremely well. The course results were astounding and the students would come out of it-- my first class at CMU I think four people in that course all changed their careers in the course to say, “This is what I want to work on.” And three of them are working directly with us still on the TSP and PSP. They just got so excited about it that they said, “It works. This is what process is.” A number of people have been working on process work on the CMM even. Once they took the PSP course, they concluded that, “Hey, now I understand it. This is really what it is.” And, sort of, that’s what it did. You really learn what a process is and why it’s helpful when you actually use it personally to do your own work.

Booch: So thinking of other people around this time, did you have any connections with people like Walker Royce’s dad, Winn Royce who wrote some of the classic papers in waterfall life-cycle or for that matter, the work that Filipe – I forget his last name – was doing starting around that time for what became the rational process. Did you have connections with any of those folks?

Humphrey: Yeah, I knew Winn Royce. He’d been-- I’d been involved in a number of things with the SEI and of course, Larry brought him in periodically. A marvelous guy, a wonderful guy. So I got to know him quite well. Of course, I got to know Barry Boehm very well and worked with him and yeah, there’s a whole list of people. I’ve been involved with just about everybody in the community one way or another, so I got an awful lot-- I never got to know the Rational folks but I did get to know an awful lot of people when I got involved with him, so I’ve got an enormous array of people that I run into that I, sort of, know.

And there are a lot of people that I know that I don’t know, if you know what I mean. They say, “Well, I met you here when I--“ I give an enormous number of talks, I did. I’m backing out now but I’ve given an enormous number of talks over the years and, of course, I’ve written so much stuff but I’ve gotten to know an awful lot of people and it’s been a very rewarding period.

Using the PSP

So in any event, we did the PSP. It was going along fine but one of the things that hit me immediately was that I would go back as my question was, how does it work on your project? I discovered people weren’t using it on the job. They finished the PSP course, they were all excited about it, it worked for them but they didn’t use it on the job. I was wondering, “Well, heck, this improves the way you work. Why don’t you use it?” And the reaction I typically got was, “I can’t.” And I concluded the reason they couldn’t was that their managers weren’t supporting it, the manager pushed them to get into test. They were planning, it wasn’t something they were supposed to do. They weren’t-- all this measurement stuff, what do you need that for?

And so all of the stuff we were doing, the quality stuff they would use, it didn’t-- we didn’t have tool support for it. Lots and lots of issues. Jim Over was putting together some early tools to support the PSP course which was helpful. It was great. We probably couldn’t have done it without that. So-- but he based them on Excel and it was fine-- worked fine but it wasn’t connected into the environment and a whole lot of stuff like that. So people weren’t able to use it. Basically working alone doing disciplined work by yourself without any support is very difficult. I was just able to do it because I was committed but I-- and I didn’t have business pressures on me. But that was it. So before I move on to the TSP and vision for the future, let me step back to the Ada design language issue and then we’ll take a break for the day, okay?

Booch: Sounds good. You bet. And I'd like to at the end of the day let's talk about what we'll cover next because I've written a few notes on things but yes, press on.

IBM Tool Development

Humphrey: Okay. When I was at IBM, this is again in my process job, I was in charge of the tool work. We had tool development work in all the labs and a bunch of things came up which were fascinating. We had this steering committee. I remember Bob Balzer and I remember Lee Osterweil and they'd review our stuff and so we had central funding for the software tool development for the company. And that was-- I had the funds and we'd basically doled them out and what we were looking at was the payoff. And we did some studies on that and found that basically none of the tools brought productivity or quality up more than 5% to 10%. You weren't getting big improvements with any tools and so one of the key things we were hitting was how do you improve quality without-- just using tools wouldn't do it. They wouldn't improve quality and productivity and all that kind of stuff. And so we concluded the proposal was that you ought to be using design languages.

So instead of just writing stuff you ought to be using some kind of a-- and the idea was a pseudocode-like language that was much more abbreviated. It was easy and quick to write and the proposal was that we ought to base it on Ada and so we had a bunch of guys put together a proposal for an Ada design language and the view was that we weren't going to up and build new IBM language like we'd tried with PL/1.

It didn't fly and so we were going to build off the Ada standard language and so that's what we did and we started to put together a design language. And people were, sort of, agreeing to use it about the time I retired. It subsequently died but the idea of using a design language did catch on and I remember the group in Boeblingen in Germany, we also had an effort we'd started on reuse because we had given them the funding for that. And so we funded this effort in Boeblingen but the effort was basically to build a parts factory and the support system and so what they did, they actually designed standard parts. They analyzed OS 360 to see what kind of stuff was standard stuff that you could pull out and make standard parts and then they rebuilt them themselves. They designed the part, essentially a sealed, unchangeable part that you could use and if people wanted to modify them they had to get a hold of Boeblingen and submit a change request.

And so they put that together and they put together a list of standard parts and they basically discovered that they needed essentially engineers-- support engineers in various labs to support this stuff. But they put that together and I don't know if it ever lasted but after a while, when I was at SEI, I was over there and visited them in Boeblingen and they had quite a few parts out there and they were-- it was doing quite a job. They had a fair amount of code in use and they hadn't had a single defect reported in all this code they'd shipped and they had used a very careful design language, careful reviews and inspections so they were able to produce high volume code without error and it was very impressive.

And so I was convinced that it was possible to get very high quality code that used really good practices. So I don't know that we used any-- I don't think it lasted and of course the whole object-oriented design and programming has really basically replaced a lot of that. That's where we were are and that was it, so we'll come back tomorrow and talk about next steps.

Booch: Yes, I'm happy to do so. Let me stop the recording.

Day 3 AM, June 19, 2009

Booch: And we're back again. This is Grady Booch with Watts Humphrey on a fine morning of Friday, June 19. I guess I should put the date because I haven't said that yet, 2009. And we're picking up, I guess, the third day of our interviews. And Watts, we were last talking around the timeframe when you were named a Fellow of the SEI but I believe you wanted to go back in time a little bit and talk about some of the situations at the beginning of the SEI itself, so I'll turn it over to you.

Humphrey: Great. Yes, a number of things that occurred to me, I sleep on these and things come popping back up, that I had not mentioned. When I was at IBM, my last job, I ran the software quality and process group. And I think I mentioned that I set up an advisory board to come in and advise us on a lot of the stuff we were doing. I wanted some academics and I wasn't quite sure who to get. And so I got a hold of a guy in our federal systems division. His name was Neil Eastman. I had heard that he had contacts and he would have some good ideas. So I got a hold of Neil and got to know him and he recommended that we get three professors, particularly Bob Balzer and Lee Osterweil and another gentlemen whose name I've forgotten.

And so we brought them in as an advisory group and they weren't people who were going to tell us what to do but they were people I wanted to have essentially review what we were doing, comment on it, ask pricing questions, hold periodic reviews with them, advisory board meetings, and just kind of stimulate the IBM folks. I had found and it's true not just in IBM but true just about everywhere, that people get tunnel vision. They're sort of there. And one of the issues we ran into at IBM and I see it in many other companies, IBM was big enough and we had somebody somewhere in the company who you could identify as a leading expert on a subject and you didn't need to go outside the company to find anything.

And as a consequence you get this very inbred behavior that you begin to think there's nothing else in the world except what we're doing in our company. And that I knew wasn't true having worked elsewhere. And so Neil was very helpful and we set up this committee. Neil wasn't on it but we had some other people on it. And it turned out coincidentally Neil Eastman ended up being the chairman of a committee for the Department of Defense to figure out what to do about software. And they recommended that a software engineering institute be established. And so Neil was sort of behind that. So I knew Neil earlier than that. And it was a sort of interesting connection. And so I've got a copy of the Eastman report here somewhere in my file. But that was what started the whole thing off.

The Speak Out Article

Now that report-- well let me make another comment. Before I joined the SEI I was still at IBM. I was on the editorial board of the *IEEE Spectrum* magazine. And one of the things we decided to initiate was a column called "Speak Out." I wrote one of the first Speak Outs. And it was in response to the issues you may recall back in those days Reagan was in place and they wanted to put this program together called SDI, popularly called Star Wars. And there was this big flap about it. And Dave Parnas, for instance, had argued that it couldn't be programmed. And I disagreed with that and this was a column about why I disagreed.

Booch: In fact, wasn't this around the time Dave was-- if I recall, Dave was invited to be on some advisory council for Star Wars and after he got into it he wrote a fairly strong letter saying, "I resign because what you guys are doing," and I'm paraphrasing, "you can't do." And he went on. So you wrote this in reaction to that, in effect?

Humphrey: Correct.

Booch: Great. That was a pretty dramatic time. I remember his message.

Humphrey: Oh yes. And so I wrote it. See what Dave had said was that you couldn't program it. And I disagreed violently with that. Later it turned out I was at a conference in Edmonton, Canada where Dave Parnas was. Bob Balzer was there as well it turned out. But I ran into Dave. And he said, "I understand you disagree with me." And I said, "Yes, as a matter of fact I think I do. How about let's go and have dinner and discuss it." So we did. And it turned out we were in total agreement. His position was that the system design wasn't possible. That you literally couldn't do it. It was the decoy problems all of the other things. That this thing was not technically feasible. And I didn't have any debate about that.

But the way he had ended up wording it was that you couldn't program it. And so that got misinterpreted. And my contention in the Speak Out was that if he was talking about a quality problem, you can't build high enough quality or high enough performance or good enough software to do the job, my position was if you could design the system and specify the software, we could build it. And so we ended up, I think, pretty much together. Marvelous guy, by the way. I don't know if you know Dave?

Booch: Oh yes, he and I have had a number of conversations over the years.

Humphrey: Yes. Well, I was on the IEEE committee that was trying to decide what were the most significant contributions in the last 50 years, the contributors.

Booch: Yes.

Humphrey: And we ended up with picking-- we basically concluded there were two that were fundamentally different but they're both very important and one was Dave Parnas. I was, in fact, deeply honored to be on that and able to participate in getting him named. It was just great.

Booch: Just a personal story, I think for a while Dave was a Fellow at the University of Limerick in Ireland and then came back to North America. Rumor had it and I've not gotten it confirmed is that he had some sort of cancer and was not doing well. The last I heard from him we had an e-mail interchange probably three or four months ago. He seemed to be doing fine then but I've not gotten any further information about him.

Humphrey: Well, I haven't talked to him in years. I really don't know him at all well, but if you do contact him, please give him my regards.

Booch: I will absolutely give him your regards, I certainly will.

Humphrey: Okay. Well, in any event the Speak Out I wrote, I've got it here. I can fax it to you. You may want to put it in. I don't know if I've got an electronic copy but I would certainly include it in the stuff I send to the history museum [Computer History Museum].

Booch: Absolutely. For the purpose of the interview, I think it's sufficient to bookmark that it's there and we can simply reference the physical copy that exists somewhere for it.

Humphrey: Right. It was *IEEE Spectrum*, April 1986.

Booch: Perfect citation.

Humphrey: Okay. So that's that. And what it really did which is kind of exciting, remember now, this in '86 before I got to the SEI, before we did the CMM, PSP, or TSP, everything. And basically, I said, I talked about many issues, and the question is-- they were basically saying you know you can't build it because you can't test it and all of this sort of thing. And I was talking about development discipline. I said the need was for a disciplined programming group. That's basically what I was talking about, and I'm saying the development of disciplined programming groups involves several phases.

Initially, you must learn to manage costs and schedules which begins with a strong commitment discipline and a rigorous estimating and scheduling process. The people who will do the work prepare their own estimates with the help of professional estimating groups who provide formal documentation, et cetera. So I go through that. Then I say the next foundation is project control and I go through that. And then I say when initially applied these will produce rapid improvement. Of course, that's what we did with the CMM. And then once schedules and costs are under control you reach a plateau where you need to make major changes for further progress.

And now I get into quality and you need to learn how to manage quality and I go through that. And then I talk about doing that and to do it properly you need to build personal disciplines and then you need to build the team discipline. And what's interesting is this document essentially describes what I've been doing for the last 23 years which is kind of amazing. I was so surprised, my team was surprised. I pulled out a copy and gave it to them a few years ago. And we kind of looked it over and said, my Lord that's exactly what we've been doing ever since. So that was part of my outrageous commitment in 1986, my vision for what I wanted to do. And I was kind of surprised when I looked at that. But that was sort of the foundation, the description of what I intended to do. Okay.

Moving to the SEI

So moving on from that I went to the SEI from IBM. And I told you about Eric Bloch and that sort of thing. The guy that was the director, the first director of the SEI was named John Manley. A very nice guy. I thoroughly enjoyed him. We had a wonderful dinner with him and his wife when I was being interviewed. And so I liked John. I didn't know him a lot because he left before I got there but he wasn't a good fit for the job. But he was a very nice guy and we had a wonderful understanding before I got there in any event but he was gone when I got there.

Booch: Not a good fit in terms of just managerial issues, or technical leadership or …

Humphrey: Yes, somehow managing the technologists and that sort of thing, he didn't-- they didn't work. They generated sparks.

Booch: Do you recall how big the SEI was at the time you joined them?

Humphrey: It was quite small. Remember when I got there-- we're going to talk about that a little bit. Because when I first got there I think Larry Druffel was running the search committee and nobody was-- we set up sort of an acting director. I think it may have been-- I'm not sure whether it was Nico Haberman or Angel Jordan but they were temporarily acting as director for a very brief period before, Lord I just went blank on his name. You mentioned his name before the guy who ran it for many years after I got there. I don't know why I go blank on his name. You know the guy that left there and went to Carolina.

W1: Carolina, yes. Larry Druffel.

Humphrey: Larry Druffel, that's it. And Larry Druffel took over very shortly after I got there. And one of the problems Larry had was they didn't have much of an administrative operation. We were in a special building over on Aiken Street in Pittsburgh. By the way when Barbara went to look for a house for us in Pittsburgh, I was still working at IBM and she went and found a house to buy and I told her to buy it. So I hadn't even seen it when we bought the house. Her criteria was that it had to be within five minutes of where I was going to work. I had been commuting for over an hour each way for the last four years. And so we had a lovely house right there in Shady Side. But the SEI was were in this old brick building and it was a nice place, but I think we must have had 40 or 50 people. We had a lot of excess funds. We couldn't expand it. We didn't have enough people. We were trying to hire and build a staff and all of that sort of thing. And so Larry-- I didn't have a job if you remember.

Booch: Right. As I recall, the gentleman that had hired you he was out the door about the time you got there.

Humphrey: Yes. So I arrived there with a totally undefined portfolio. And they asked me if I would take a secretary who was the ex-director's secretary, John Manley's secretary, a lovely lady. And they didn't have any place to put her. And so I arrived, I had nobody working for me. This is the first time I had had nobody working for me since college and it was a marvelous feeling. All of a sudden, I wasn't administering anything. And so I was able to start thinking and putting together my own ideas. Well, that didn't last very long because Larry right away needed somehow to pull together an awful lot of pieces, the administrative parts, finance of the SEI. And he asked me if I'd run that. And I told him I would temporarily do it-- I would do it on two conditions. One it was temporary, I'd be working out of that. We'd get somebody in those jobs quickly. And second, I could then form my own process department. And so he bought that. And so fortunately, I was now in the position of setting up and distributing funds, laying out budgets, running the whole group, getting this place going. And I guess I basically was able to ensure that we had a properly funded process group. So that was a nice dividend out of that. So we did that, and we got that thing all set up. And this is while I was going ahead and starting with Bill Sweet and we were starting on the CMM thing I had mentioned before with the folks up at MITRE in Boston. So I was doing that while running the administrative stuff. And I had an office in the top floor of this building on Aiken Street. And Mary Shaw turned out to have an office right there on the same floor.

Booch: What a wonderful lady she is. Where was Dave Garland around this time too?

Humphrey: I didn't see Dave. Dave wasn't there. He wasn't there at the Institute; at least if he was I didn't have any interaction with him. But I remember Mary and I would just have lots of debates kind of informal discussions and we were on a totally different wavelength. I mean the process that she did not view as any kind of rational science or knowledge. She was interested in stuff that was sort of hard knowledge. And so we had lots of debates about that. And at one point I don't know if she proposed it or I did, but she was having a lot of trouble getting her work done because people constantly wanted to come meet with her.

So she and I established a schedule where we would have one or two afternoons a week where Mary Shaw and Watts Humphrey were meeting. Period. And we could not be interrupted. And we were meeting by ourselves. We weren't together, but both of our calendars were blocked. And so we were both able to get lots of work done that way it was marvelous. So my Mary Shaw meetings were very productive. So I thoroughly enjoyed Mary although we disagreed on a lot of stuff. Another thing that happened in there was quite early. I remember, I read an article in *Business Week* magazine. I think this could very well have been in '86 or '87 but the records would show. I don't remember the date. Remember the Ashton-Tate company?

Booch: Yes, I do. They were one of the early PC companies. Absolutely.

The Aston Tate Story

Humphrey: What's interesting is Ashton Tate, I think, was second in size to, I don't know, Microsoft or somebody but it was one of the big three companies to the software business. And it had this dBASE I program which had really been an enormous success. And they then announced a dBASE IV to replace it, upgrade it, to compete with Oracle and others to stay in business. And so they did that and this *Business Week* article was about the problems they were having with dBASE IV because when they developed dBASE IV it was running late, all kinds of problems for them, they weren't able to get it out of test. So instead of finishing test, they stopped the test which is a normal way they do it in a lot of places. And of course, how do you know when you're finished testing? You don't which is one of the fundamental problems. But in any event they stopped testing and shipped it. And the basic position was "we'll fix it later." Well, it turned out with the database system they had a number of defects that were actually destroying people's databases. That's not something you can tolerate. And so they rather quickly ran into serious problems. They had to pull the program and go rebuild it to reship it. And this was going on when I read this *Business Week* article and they had no idea when they'd get done. They were under enormous pressure, all kinds of things. So I called the executive offices of Ashton-Tate and asked to talk to the assistant of the CEO and I talked to him.

Booch: And their CEO at the time I think was George Tate was it not?

Humphrey: No, it was Ed Esber.

Booch: Okay. I see it, Ed Esber, yes.

Humphrey: Was it Esper?

Booch: E-S-B-E-R is the name I found.

Humphrey: E-S-B-E-R.

Booch: Yes.

Humphrey: Okay. I had E-S-P-E-R but it's B-E-R, okay. But in any event so I told him and you remember I told you before about defect clustering and the strategy we had worked out at IBM.

Booch: Right.

Humphrey: And so basically I told him that I had a strategy I think would help him them accelerate getting this thing out the door and I'd be happy to come talk to him about it. I asked them to pay my airfare which they agreed. So I flew out. I've forgotten where their office was. It was probably San Jose or some place near there. So I flew out there and arrived. And the CEO Ed was there and he had all of this top executives there except one.

Booch: Torrance, California by the way is where their headquarters were.

Humphrey: Is that near San Francisco?

Booch: Yes, it is.

Humphrey: So there was one executive who couldn't make it, guess who that was? The development VP. Of course, he didn't show up. So we talked briefly about what they were doing and their status and I described the strategy that we had worked out that had been enormously successful. And Ed thought that sounded very interesting. So how long will it take? This is like in October. Maybe it was a year later maybe it was October '87 something like that, but it was very early. And I said it'll take about three months. It'll take about a month to get a team of people really trained to do good inspections. You've got to get some data on your testing. Then you've got to take probably a couple of months to go through at least the 10 percent of your worst modules, inspect them and fix them and then you can probably get the quality pretty well fixed up. I would guess it would probably be January or February. And Ed's reaction was, "We can't do it. We've got to ship in November." And I said, "You won't ship in November unless you do something like this." He said, "Oh no, no, we're going to ship in November and that's what everybody says, et cetera, et cetera." So they said no. Well, as you know they didn't ship in November. They didn't ship the following April and they went out of business the following spring. And I see that message all of the time. "What has to be will be" is sort of the feeling and it wasn't. It just didn't happen period. And so it was a real frustration for me how hard it was to get people to actually buy the kind of stuff that they have to do to the job right.

Booch: So I'm hearing you say that a common theme you're seeing is that people are happy to do it wrong and invest the time to fix it, which sometimes never works, as opposed to investing the time to get it right the first time.

Humphrey: Well, it's not even quite like that. What it means is they have this optimistic feeling that they can fix it. Like these people, the view, and I see this at Microsoft, in particular, and other companies, too, "if we could just get the right tool we can do it." And for instance in the security business I see that all of the time, the buffer overflows. The security problems on the Internet, over 95 percent of them are due to

standard garden variety defects in the code. It's amazing. A lot of them are defects that most programmers won't recognize. A lot of them can't identify what a buffer overflow potential is. But Microsoft has a wonderful tool for identifying buffer overflows and going through and making sure they don't cause a problem with the system and identifying them so you can fix them.

So when I talked to some people there that had a little data about that tool they had I said, well what percentage of the problems does it find? About 20 percent. I mean you're going to hang your life on something that finds 20 percent of the problems? So that's what we're doing. People aren't really looking at the data. They don't understand it. They're not looking at it scientifically. And that I think is a serious problem. So I ran into that with those guys too. They wouldn't listen to the facts. They couldn't believe them. They would rather believe their own mythology than focus on what reality really is. And that's basically I think what the whole community has been doing. We're not facing reality. Reality says testing forever won't produce quality products - we know that. We know that for every other technological field but we still struggle with that with software and that's where the whole academic community is saying "We can improve testing." There was a big report to the president on what we had to do with software quality problems. It was all about that. More testing. Better testing. It's insane. In any event, I'll get off that soapbox again for a while.

Nico Haberman and the SEI Strategy

Now, let me move back to something that happened right after, fairly early on, you remember now there were a bunch of techies there at the SEI. I mean good people but they just weren't on my wavelength. And Nico Haberman who had been the computer science chair or dean, and I think was at this time, and he was the guy who I think had actually conceived the whole idea for CMU proposing that the SEI be there. And he was the guy that basically put together their proposal. And he was a marvelous guy, a Dutchman. Highly revered. Unfortunately, he died prematurely. He had a heart attack running on the beach I think. I'm trying to remember what year that was. It was in 1994. Remember because we were in Sorrento at the ICSE meeting-- actually the date of our 40th wedding anniversary and so it would have been in May of 1994. And they had a memorial service there for Nico. It was a terrible loss. And he was a young man. He was a good deal younger than I was.

In any event, Larry Druffel called a strategy meeting shortly after he became director. It was a little bit after because we had gotten far enough along to put together the ideas for a maturity model. And remember I mentioned that I had sort of dreamed up this five levels and sort of worked off that "quality is free" idea and that sort of thing. And we put together the five levels with these questions we put together with the people from MITRE. And so we arrived at this strategy meeting with all of the techie people and Nico was invited. And so during the meeting, we broke into about three subgroups, working groups, to put together what we thought the SEI strategy ought to be and we were going to have these three presentations at the end. Well, by great good fortune I happened to be on Nico's group. And we were going through a discussion and they came to me and I got up and I described this five level model and how you use it for improving software development work and that sort of thing. And the guys there were panning it, right? Well, what's that for? Why do I want to do that? And Nico interrupted and said, "This is why we formed the SEI." He said that's it, "That's marvelous." And I'll tell you without his backing I don't think I would have been able to get it done at the SEI. I mean he could see it. It was exactly what he was after. So I thought that this man had vision and he really did, he had marvelous vision, a wonderful guy and I was really sad that we lost him so early. But in any event, so he played a very powerful hand in forming the SEI and in putting us on the path that we've been on ever since. Because without that we would never have had a process program that went anywhere at all. Okay.

Booch: Very well.

Humphrey: Well, now, let's shift gears, because I think I've pretty much caught all the way up to where I was made a fellow and Barbara had decided that she wanted to live in Florida in our house and so we moved. And I had gotten this idea that to look at improving programming we had to start at the bottom. And this is sort of what came out of my education as a nuclear physicist, originally, it's sort of that you need to understand the base forces and the data and what's happening and the foundation underneath everything before you can begin to build a structure. Which seems logical but it's hard to get people to buy it. But in any event, so that's what I went to do. So I decided, and Larry agreed. So I just went down. Actually I started writing monthly reports as a follow up because I was writing programs too and I was kind of bored. You know you're working alone and I used to have this feeling, I wasn't bored in a sense, I was thoroughly enjoying it but I had a feeling I wasn't making any progress. It's very hard to get a feeling of accomplishing anything when you're all alone and there's nothing going on. You sort of see no motion if you know what I mean.

And so Barbara rather wisely suggested, why don't you put together a monthly report? And so I did. So I started writing a Fellow report and I wrote one every month and I've done it ever since. And I've got them. That's part of what I'll send to the museum, my monthly reports. So I wrote about the meetings I had and the trips I made and what I'd done. And I was really quite surprised when I started looking at the- at the end of the month that I had a whole bunch of items I'd actually accomplished. I'd been off and done this and I'd been given- asked to come give the keynote speech at some place and I had some conference calls with people. And so I was actually accomplishing a lot more than I realized. And so that all of a sudden was a great feeling of more self-worth out of it. It's very helpful to do that. And that's one of the reasons I kept it up. And so I've been doing that and I'd send copies to Larry and a lot of people have asked for copies and so I was sending copies to a distribution list at SEI but since my cancer, I basically stopped. My activities are so limited and I'm writing it but not widely distributing it anymore.

Booch: Well, let's consider this interview here one of your extended reports.

Measuring Myself

Humphrey: Right, okay. In any event, so I was writing the PSP programs and a whole bunch of things came up as I was doing it. One of the questions was "exactly what do you want to make?" And I think I measured- I mentioned this a little bit earlier. I decided I wanted to get really basic measures. People get all confused about what measures are and I wanted things that were auditable. And think about it this way: think about a measurement system that's scalable. Can you think of one that really is scalable from the smallest to the biggest business or operation?

Booch: Well, source lines of code comes to mind.

Humphrey: Well, yes, it is an auditable measurement system, and source lines of code is a measure of size. But a widely-used measure that is scalable from the smallest to the biggest business is cost accounting. And you use- people in a mom and pop shop, individuals, and I took cost accounting in my MBA which was very helpful, but they use it and it scales up to the largest corporation. Everybody uses that and the reason is they're working from auditable real data. And you can count on the data. You can actually build it and define it. It's firm, it's clear. They say that the data don't lie. I mean liars can figure but

the figures don't lie. You know what I mean? And so that's what I tried to put together: an auditable foundation of data. And what I've discovered was that to do that I had to have really well defined data.

And then I realized that the pretty basic data items that I concluded I needed were size, time and defects. And that's the size of everything you build, the time you spend doing every action, whatever you're doing, every task, and the defects you find at every step. Now to do that, to define that, however, that means you've got to define the steps of your job. So you must have a defined process, you must connect the tasks that you're doing to your process, and then you connect the data and the product so the tasks have to connect to the products and to the process.

And so it's a big interconnected system. But if it's auditable and if you have real sound data under it, what we have discovered is that I can put together a report on a program I write of 100 lines of code and I can look at it and we can do that identical report with that information in it and we can scale it up to a system of several million lines of code. The identical numbers. Now as a matter of fact, you can then look at it and you can say, well what's this mean and you can bore back down because it's now constructed from an auditable base. And you can go find all the data for all the parts to justify it: the defects, the time spent, what happened, how big this was, the changes, all the activity-- I mean it's amazing what you can get out of this stuff.

Booch: So the first two things you mentioned are fairly quantifiable but defects seem to need to be something that there's a spectrum of different kinds of defects. So in your early incarnation of this how did you characterize what a defect was and did you have a sense for different classes of defects and if that would weigh things?

Humphrey: Yes, yes. Well that was, of course, the first question: “what is a defect?” Lots of debates with that and then you try to get an agreement with the lawyers as to what a defect is but don't bother because you'll never get there because they are all looking for blame. But the issue I found was if you focus on it in terms of actual actions, what you have to do, the programmers have no trouble identifying defects. And I didn't either, it's something I have to fix period. It was very straightforward. Now I had to define them and you have got to be very careful defining defects and because people want to confuse the cause with the actual nature of the defect.

And so a defect, you know, may cause a buffer overflow and there may be things that it does and the results and the problems they cause but it is in fact an error in a loop. Or it is, you know, some particular error that you've got in the code. And they could be trivial errors. I remember there was one defect I remember. Some guys were teaching the PSP at DEC, and one of the guys called me from there, they'd been to PSP training, they never went very far with it, which was too bad because as you know DEC got bought and things got moved and management changed and very hard to maintain this improvement in the face of that dynamic.

Booch: And their artifacts became eventually the Computer History Museum.

Humphrey: But in any event, DEC had this non-stop computing system where they would have- they could put two or more processors together and if one processor had a problem, a second one or a third one would pick up and they had this system and that was a marvelous system and it was actually guaranteed to run without failure. The problem was it was failing. And so the guys called me about it. They told me it was- it was really quite an amazing story because they had struggled with this thing. It

didn't happen very often but when it hit it basically- all the connected machines locked up, bam, and that was it. And they had to go and shut the whole thing down and restart. And I'll tell you that was a disaster. And so this guy called me, I think his name was Goldman, matter of fact, interesting guy, he worked with Howie Dow who's the guy that I had known at SEI that taught them the PSP. And he called me, he said what had happened was that they were just about-- I'm going to back up.

When this problem finally became so severe that this manager decided he had to take his two best engineers and assign them full time to find and fix this problem. And so they did. And they worked for two to three months before they finally figured out what it was- and they finally did. And before they fixed it, the manager said, "Oh hold now, because where that defect is is code that they're just about to do an inspection on for a revision of the system. So let's just participate in the inspection and see if they find it." So he did. And he called me to tell me the results.

The guys had gone through the inspection, took them a couple- several hours. Remember now, the two best programmers spent three months on this, the team- the inspection team spent a couple of hours going through the code and then they were going through the defects and one of them pointed out this defect. And so the manager asked them, he said, "What kind of trouble do you think that would cause?" And the guys thought it would be kind of an annoyance but they'd probably find it in test and this sort of thing. And he asked how hard they thought it would be to find. You know, it'd probably take a little while in testing. And then he told them the story.

Well, it turned out it was a trivial little error and they could fix it in a few minutes and they found it real quick with an inspection when they really had a team focused on it. And that's what we're telling engineers they need to do. They need to do personal reviews; they need to do team inspections. Don't count on testing because some of these trivial little problems have enormous consequences and that's what this was. It was a very simple little problem. What it caused to happen is that one- it was basically the trigger problem where one computer would actually end up needing feedback from another one and the other one would end up getting in a loop where it needed feedback and the two of them would both essentially wait for feedback from the other one. So they'd basically hang. And it was a simple defect in terms of making sure the hang wouldn't occur. And it was a trivial one but as you know, those trivial errors can cause enormous trouble.

So the issue here is to separate out what the defect is. In this case it could be a- you know, a bit that's set wrong or it could be an overlooked loop closure or it could be an off-by-one error. It could be- you know, you name it. The consequences are another thing and the causes are another thing. So the defect classifications that we have for the PSP simply relate to what the defect itself is: what you've got to fix. And if people want to go and analyze causes and all that, the defect data are very helpful and can be used. But when the engineers are just looking at what it is and what's got to be changed, they have no trouble identifying fixes and we've got data now on something over 30,000 programs written for the PSP. And every PSP program the guys record every defect. And they don't have any trouble. No one argues about it, you just say here's all you do and everybody does it.

And so while people may say that you can't actually count the defects, we have no trouble. And I have thousands of engineers' defect data. Now we're doing it with no trouble, no one debates it, no one argues about it so it's a lot of smoke. And the reason I say it's smoke is because I'm talking to people who haven't actually tried to do it themselves. So if you sat down and were to track all the defects that you find in your programs, when you review them, when you compile them, when you test them, when you build

them, whatever you do, track every one, look at it, figure out what it takes to fix it, track it. No big deal. So, that answer your question?

Booch: Absolutely. It does very much, thank you.

Humphrey: Okay. But the point here- and that's why I wanted to sit down and write the PSP programs because I was getting all this smoke about stuff you couldn't do and why, and this and that, and so I decided to just try to do it. Here's what I thought we ought to do so I was trying to act like a CMM level 5 organization of one person. I was trying to do everything I thought was needed and put the whole thing together and I did. And it was extraordinary. So that's what the PSP was all about.

Booch: And while you were doing this, who were you primarily collaborating with? Or was this springing largely from just your mind and work.

Humphrey: It was me. It just did it myself. I had- I was working on statistics because I needed- I discovered a lot on my analysis work I needed simple statistics and so I worked through a statistics book, a graduate text on statistics. I remember it was kind of funny. Barbara and I went on a trip to Turkey, went on a tour about three or four weeks. Marvelous trip by the way. Our youngest son basically told us once that his favorite country in the world was Turkey so we decided to take an Elderhostel tour to Turkey.

And I took along my statistics textbook and so I spent a fair amount of the trip just evenings and spare time just going through that. Although we did see Capadocea and lots of other wonderful places. But so I remember going through that and learning statistics. And I had to learn that and so I had some people in the statistics department at CMU that were very helpful, John Lahosky and others, I had the people at the SEI help me with programming problems. I hadn't programmed in years, I didn't know the modern systems and so I'd call folks, Jim Over and others and say, "Well, how do you do this?" And so I had all kinds of people help.

Initial PSP Use

So a lot of people were very helpful with my programming problems. But basically coming up with the PSP and the PSP idea and how to do it, by and large pretty much just about everybody I was working with at SEI and elsewhere thought I was kind of a nut. I mean “what in the world is the point of doing that?” was sort of the reaction that I was getting. And remember I’d gone through and written the programs and then I gave a talk down in Texas to the people at TI. I did talks to a bunch of places trying to get people interested in trying the PSP. A couple of graduate students at CMU, who were PhD students, they expressed interest in it and so I got them copies of the process and said here why don't you write- use that to write some programs and they couldn't. I mean they used bits of it that they thought would work and they didn't bother with the others, they didn't have the discipline, they didn't have the data. I mean they were bright guys but they really didn't have the dedication I did. And, of course, without a tool it's not that hard to gather the data but it's hard to get programmers to gather data without some kind of support tool.

So in any event- and I- even there was a group- a Siemens Research lab in Princeton and one of the engineers there had heard about what I was doing, he called me, he was interested. So I went down and talked to them. And they were very interested. They wanted to do it, I talked to the team, they were

excited about it. The director of the lab actually told the team that his top priority was to have them do it this way and that the schedule and everything else was secondary. And I thought this was a level of management support I really needed. The guy was great, I thought it was marvelous.

And so I went home all encouraged that I'll get these guys to actually use it. So I'd given them the process, I'd described it to them. And so I kept calling like every week: where do you stand, how are you doing? Oh we're doing a prototype here, we don't have time to do it yet. And they never did. They never got to it. They were in too big a hurry to do other stuff. What was interesting was as part of that I'd actually taken time to sit down with each of the engineers on the team. There were five- I think five engineers and they had a process guy there but he was sort of a technology guy but he was sort of acting as the process guy I guess. But in any event, he and I would sit down and interview each of the engineers individually on the team. And what I did was I wanted to find out what process are you using personally. And they didn't know how to answer that question so I'd kind of lead them through it.

I said, okay, when you get an assignment to write a module of code, say 2,000 lines of code, what do you get? So they'd tell me and then say here's how I'd start. I'd say well what do you do next? So they'd start to describe that. And I said well how about, you know-- So they would describe what they did and I'd lead them step by step and I wrote down on the board what they were saying. And every so often they'd say here's what they did next. And I'd say, "Well don't you have to do this too?" And so they, "Oh yeah, well I forgot that." And so they- but we built the process for each of these five people. I don't know if it's in my notebooks or not. But I wrote it on the board. I'd write it where they could see it. But in any event, they were all, the five of them on one team, totally different. It was amazing. I mean one young guy came right out of school, seemed like a real sharp guy, but he basically started coding in the middle and he didn't do any design work at all. He just sort of had this idea of what it was and how it was going to work and so he started coding in the middle of this big thing and he'd sort of build it.

And at the other extreme, there was an engineer, I think he was actually from Germany, and he told me that he went through his process which was a marvelous process, very disciplined, he didn't have data but he really did a very careful job, and he said he'd never had a defect found in any of his programs. I mean this guy was proud of what he was doing. He was producing really masterpieces of programming. And I thought to myself, we've got these masterpieces going in with this junk. It was just staggering to me that we had this enormous range of the quality of the code going into one system these people were building. And it was sad. And my point is we see that everywhere. Nobody knows who's using what process or what quality they're producing. So we get beautiful code by some people and junk by others and the junk will kill the program. Just kill it. And so they run into all these problems, months of testing. Well, their product they never could get it to work. And the reason is the junk, there was just too much of it, and they could never get it fixed. So that- that's kind of sad.

And so the individual process is a critical piece of this and that's why I went all the way to the PSP. And so, as I say, I never could find anybody who would use it and so I was really very frustrated. And then I was in- I think I may have mentioned I was at a conference in Berlin, a process conference where Peter Feiler and I gave a paper on terminology and process. It's one of the technical reports we got. But while I was there, I talked to Dr. Mahdevi who is the professor who suggested that I teach a course and I wrote that textbook I mentioned. And I had a marvelous time because I was going to these various symphonies and they- I remember it was interesting sitting in the audience watching a symphony play because, as I said, I went to a concert every night when I was in Berlin, it was a thrilling experience. I was there for what must have been almost a week and they have such a rich place, their symphony hall is gorgeous, you know, round and you can sit sort of behind the symphony.

Booch: And the time you were there that's when the wall had just gone down so the dynamics was amazing.

Humphrey: Oh yes, the wall had come down, I walked over there and around it, was looking at the whole area.

Booch: And you saw the piles of rubble in East Berlin. That was amazing to see.

Defective Software

Humphrey: Oh exactly, exactly. It was a thrilling experience. But in any event I was looking at that and it's amazing, by watching how symphonies work what kind of feeling you get from the dynamics of doing beautiful work. And it was really kind of amazing. And that's the software issue. We really need symphonic teams. The hacker business is so sad because, just like in a symphony, any individual instrument can destroy the whole effect and that's exactly true of software. Any individual piece of code can destroy the whole thing. And that's the problem. I'm trying to remember the name of the medical instrument, remember that killed a bunch of people, the Therac-25 machine. And that was a trivial error in an error recovery program. I mean it wasn't something that normally gets used. It was off in the side somewhere. I remember one example- an interesting sideline, earlier at IBM on OS 360 we had performance problems.

Booch: The Therac-25, that was the machine.

Humphrey: That's what it was, the Therac-25. And that was an error recovery program that had a defect in it and it missed getting a reset and I believe killed half a dozen people. And so we get those problems. And exactly the same thing with the 20- with the V-22 Osprey. Do you know the story about the V-22?

Booch: I do but the people listening in might not so why don't you relay.

Humphrey: Well, on the V-22 Osprey, I actually I went out and talked to the executives of the company that built that system and I was talking about the quality of their software. They bridled, they said we have very high quality software. I said, the fact that it's killed 13 marines is a good measure of quality. They didn't buy that. But in any event, the V-22 Osprey is this tilt-wing aircraft that you can fly as a regular airplane and then as you're coming in to land you tilt the wings up and so the propellers are pointed up instead of forward and it lands as a helicopter. And- I mean it's an enormous technical achievement to build that thing.

And, of course, one of the issues that they had was what happens when the hydraulics fail while you're tilting the wing? You've got a whole hydraulic system that does that and so they put in a whole electronic backup system in case you have a hydraulic failure. An electronic backup system that will fly the airplane electronically. And that's what was in it, and it turned out in this particular case they were in a test flight with a bunch of marines in the aircraft, they were coming in to land and as they were tilting the wings to bring it in, the hydraulics failed.

And so the backup system took over, the software that controlled the electronics system took over and the software had an error in it that essentially crossed the controls. And of course a pilot can figure that

out if he's got a little time but when you're coming in to land that's a little hard and they crashed and killed everybody. And the point is- and I run into this in all kinds of things, the number of possibilities that have to be tested, and this is what the executive was telling me, "Oh, we tested it exhaustively." And I agreed they tested exhaustively but exhaustive testing won't find all the defects. People don't know that. They don't understand that. And let me branch a little piece on here. When you think about a big program, big complex system program, 2 million lines of code something like that, and you run exhaustive tests, what percentage of all the possibilities do you think you've tested? Any idea?

Booch: Oh it's going to be an embarrassingly small number probably in the less than 20, 30% would be my guess. What would you say?

Humphrey: You're way off. Way off. I asked- I typically ask people and I get back numbers 50%, 30%, that kind of thing. I asked the people at Microsoft, the Windows people, what they thought. And then we chatted about it a bit and they said about 1%.

Booch: Oh my goodness.

Humphrey: And my reaction is they're high by several orders of magnitude. And let me explain the reason why. The conditions that actually affect testing. I mean testing will only test a specific set of conditions and the conditions that will affect testing include, for instance, how many job streams are running, what the configuration is for the system at that time, all kinds of things. And also what the operator's doing, what the hardware conditions are. So you can have a hardware failure, you could have data errors, you can have operator errors, you can have just an enormous range of things. And if you make a list of all the variations, and then by the way you need different data values too. And so you got different data values. So if you go through and actually see what are the conditions-- I did this on a program with 57 lines of code that I'd written for the PSP. I went through and analyzed exactly how many test cases I'd have to run to exhaustively test it. I didn't worry about different data values yet, I assumed I would classify those and I could come up with that and I never did go back and do that. But it was like 268 test cases for 57 lines of code. I mean it's extraordinary. And that's true. So people can talk about automated testing and that sort of thing, but the number of possibilities is so extraordinary you literally couldn't do a comprehensive test in the lifetime of the universe today.

Booch: So in effect there's a combinatorial explosion due to the number of possible states.

Humphrey: Exactly. And you look at all the number of possible ways things can connect, I mean it's extraordinary. And so when people have this enormous faith in testing it's vastly misplaced. And so the quality problem is really severe. And that's the issue that I was getting at. My sense was if you didn't deal with quality exhaustively at the very beginning, at the smallest unique level of the program you will never solve the problem anywhere else. And that you can do. And so I found, and this is what I was after finding out with the PSP prior to my quality study, could I produce defect-free programming? And my contention that a program was defect-free, if I wrote the program, I had a design, I went through it, I produced a comprehensive test and if I wrote the program and I compiled it without error and I ran all the tests without error, then I figured I probably had a pretty good program. Now that was without error the first time I ran the tests. So I'm now treating testing as not a way to find defects, it's a way to verify the quality of the work I've done.

Formal Methods

Booch: A question for you with regards to the defect-free programming because my recollection is that this was around the same time that Dr. Tony Hoare was talking about the notions of formalisms in development. Edsger W. Dijkstra of course as well around that time. And their thought was, you know, really formal specifications of the programs but you were looking at an alternate view with regards to process. How did that fit in? What knowledge did you have of what they were up to?

Humphrey: I got fascinated with those- with the formalisms and it was very attractive. I took a tutorial with Tony at ICSE, I had a group working in Germany while I was in my IBM quality process job. I used to go and visit with the people particularly in Germany because people there, they were doing marvelous work. A man named Horst Remus at IBM, retired about the time I did. He introduced me a bunch of these people, and they were academics they were working with in Germany, and they had these formal design methods. I also ran into some in the Netherlands that were doing that. So I was looking at that, the Zed language, if you're familiar with that.

Booch: Very much, yes.

Humphrey: So I got interested in that, and I actually tried some of these methods on my own work, and I found it was quite helpful-- the discipline was helpful-- but I found the problem was I wasn't that familiar with the formalisms, and that I tended to make more errors when I did it that way than I did otherwise. Part of the reason was that I wasn't that good with the formalism stuff, so familiarity with the language was a real problem for me. But the other was that I then had to translate it into code, and I made a lot of errors there too. So my error rates actually went up rather significantly when I tried to do that, and I think part of it was my own problem. I think the formalism approaches will work at the individual level, and I think that's useful. One group-- I don't know if you're familiar with the people with the SPARK programs at Praxis in England. Have you heard about those?

Booch: I have. There's John. One of the early Ada guys was involved in that one.

Humphrey: Oh yeah. What they did is they took an Ada subset, and they subsetted Ada to make basically what they concluded was a precise language that didn't have all of the misunderstandings and confusions in a lot of the languages. I actually went and took their course, their SPARK course, in England.

Booch: It was John Barnes. I think that was the guy still involved with it.

Humphrey: Well, I just got an e-mail from one of the guys the other day. They've now made their whole language and toolset available open source, which is marvelous. And they're having good experience. One of their top engineers-- Rod Chapman I think is his name. I went over there and he taught me and Noopur Davis, one of our people, and we had some other people in the class. He actually taught us the SPARK method, and we wrote some programs and went through and did it. It was an eye-opener. To identify all sources of buffer overflow is a hell of a challenge, and the mathematical methods are extremely helpful in doing that. So I found that I'm very impressed with those methods; I think they're very powerful. And Rod Chapman actually came over and got trained as a PSP instructor, a TSP coach. So

he's qualified in the PSP and TSP. And Rod and I have discussed this at some length, and we both believe that the two methods are quite complementary.

They both address important but slightly different facets of the problem. And we both believe that the two methods really ought to get merged. One of the problems of course with SPARK is that it's an Ada subset language, so that's a problem. Later on, when I talk about the future, I want to talk about languages. So I agree, the formalism stuff is great. But what was amazing to me was we can get such an enormous level of improvement without it. And that's what I've found is amazing, is we go through training engineers with the PSP, just gathering data on their defects, just looking at it, analyzing defects.

The reason it's so extraordinary is that when you make a mistake-- and I tried to eliminate my mistakes, and I literally couldn't do it. I found that there was no way that I could write programs without making errors. And the reason was that I was human, contrary to popular opinion, right? But there was no way I could literally do it. Because I was human, I tended to make various kinds of errors. But the thing that was interesting about it was I tended to make a particular category of errors. So I'd make them repetitively, and what surprised me was that if I gathered data on the kinds of mistakes I had made and used that data to put together a checklist to go back and review my code to find those errors, I got damn good at it. I got to where I could find most of the errors before I even compiled my program. I didn't get to where I could find all of them. I did occasionally. But I find really good programmers do get that good.

The Yield Measure

What we've got in the PSP is the measure called "yield," and you have the yield data. What percentage of the defects do you find in compile, and what percent do you find in the testing, what percent do you find in inspection or a code review, and that sort of thing. What I've found as I've looked at the data, on code review yields-- with a lot of work, you can get up to 70 to 80 percent. I was typically getting close to 80 percent yields on my code reviews, which is pretty good. I find most people don't even bother doing a code review, so most of the defects just slip through. You find also compiler yields are typically in the 50 percent range or less. I find a surprising number where the defects that you would classify as syntax defects-- but they're not really true syntax defects. It's like a misspelled word that turns out to be a valid other word. So you make syntax defects that turn out to produce valid syntax. So it's not really a syntax defect, but it's that kind of a problem. It's a typo; it's a whatever. But it ends up producing valid code. It will compile.

So one of the things we try to convince people in the PSP course is that compilers or builders, whatever you got-- I know many people are getting away from compilers-- but whatever that tool is, its job is to try to take whatever you produce and make running code out of it. So it's not really trying to find errors; it's trying to produce code. So to count on any automated code-building tool like that to find errors for you is a mistake. Now, that isn't true of things like the SPARK tools, which are designed to find errors, that really analyze it, and they're looking at it. But that certainly is true of most of the code-building tools.

So that is really kind of a key question. You really have to focus on the quality of what the individual does. What I've found as part of that, that the various kinds of quality reviews-- I do personal reviews. I review designs; I review the code. So I produce a program, for instance, and suppose I spend an hour writing the code. I force myself to spend at least 30 minutes going over the code, going over it carefully to try to find every error. And as I said, I can find 70 to 80 percent of them, or I could when I was writing programs a lot. So people can get pretty good at it. But what's interesting is the defects I find personally when I do

reviews, I'll find them quickly, but I'll find quickly a lot of the defects that would take you forever to find in tests.

Conversely, defects that are bloody hell to find individually pop up right away in tests. Like the blue screen-- all that stuff. It doesn't work at all. Well, some of those things you completely miss,; the systemic-level stuff shows up right away in test, but you can't see it. So there are a whole lot of issues like that that I find are kind of fascinating. And the thing that's amazing to me is if you take and improve the quality of the code going into test by 10 times, you'd think you'd improve the quality coming out of test by 10 times. What's amazing is you don't. You improve it by close to 100 times. Sort of amazing.

So it takes a relatively modest investment, but we're getting extraordinary improvements in quality by really addressing the individuals and getting them conscious of the quality of their personal work, and giving them data on it so they can look at it. So that's what we did with the PSP, and that's what I've found. So I taught the PSP course at CMU, and I think I mentioned that four of the people who took the course got so excited about it that they decided to change their lifetime careers and I've got people working with me now that were in that course.

So it's a transforming experience to go through this, and we have thousands of engineers who have now done it. And with very few exceptions, they come out of this saying, "Wow. That's the way I want to work." I'd say about, oh, 60-70 percent come about with a sort of "wow" feeling. Another 30-40 percent come out with an, "Well, okay, but I'm not sure I can do it on the job" feeling. And then there's a small number of hardheads that really don't buy it. But it's kind of interesting. But in any event, so we got the PSP going, and we started to teach classes and bring people in to take courses.

I worked with Embry-Riddle University, and they gave me my very first experience with using the TSP. I put together the process and I talked to the people at Embry-Riddle who-- let me back up before I do that, because I ran into Embry-Riddle people at an SEI symposium. SEI would run a symposium every year. We'd bring in DoD people and industry people, etcetera, and we'd present our plans and what we were doing.

I was in one of these meetings; I made a presentation and I was sitting watching somebody else, and a guy came up and tapped me on the shoulder and said he wanted to talk to me. So I stepped out to see what he wanted. His name was Soheil Kajenuri. He was a professor at Embry-Riddle University, and he said, "We want to do the PSP at Embry-Riddle." No, I'm sorry, that wasn't what he was saying. He said, "We're building a master's curriculum at Embry-Riddle, and we'd like to build it around the CMM." That's what he said. So he said, "I'd like you to talk to us." I said, "Well, I'll come out and talk to you. I can't do it now, but I can come out." We picked the time.

So he was there with their department chairman, Iraj Hirmanpour, and so I met with them. So I went through what they wanted to do, and they wanted to build a curriculum about the CMM. I said, "Well, basically, doing that's fine, but you need to have the PSP in there." So I described that and all of that, and they got all excited about it and said, "Yeah, that's a good idea." So they made the PSP the first required course in the graduate program. And Soheil taught a PSP course shortly thereafter, out of my textbook. So I think the book was published by then, I believe, I don't think it was the manuscript. Although it may still have been a manuscript. This was quite early.

So it was like 1995, ’94, something like that. Some of the first people writing programs in ’93-- that’s when Howie Dow did his. I taught in early ’94. That’s right. And then my discipline with PSP book, which I taught from in January through June of ’94, I was getting the class data, and then using that data to re-do the book. So I rewrote that book that spring and summer and got the manuscript to my publisher that fall, and it was published actually in December of ’94 with a ’95 copyright. By the way, a side point, I worked with Addison-Wesley, and Peter Gordon is my editor.

Booch: Who was the editor again? You broke up just a little.

Humphrey: Peter Gordon. He was my editor on the *Managing the Software Process* book, the very beginning. He’s been my editor now on nine books. Early on, I used the PSP to write my books. And he told me, he said he’s never had an author who makes every schedule. And I have. I mean, I’ll go through it, I’ll make a plan, and I’ll meet it. Things don’t come out to be completely predictable. I end up completely rewriting books and that sort of thing. My *Winning with Software* book this just didn’t work on, so I ended up throwing it away and rewriting it three times. So I was predictable on each version. But in any event, so I had the book, Soheil taught from it, and as I say, it was working great until the people at Embry-Riddle started trying to teach local industry people. They got some companies that were interested and they were teaching there, and they had several courses at like Boeing, Motorola, and other places. And I kept going to check, and I discovered that even though they had the course, and the students came out all excited about it, and they’d start using it on the job, it died out very quickly.

So PSP use did not persist. There are various reasons, but fundamentally, the reason was that the engineers said-- I think I mentioned this before-- that they were working alone. I mean, they were trying to do disciplined work. The analogy I use is trying to practice to be a concert pianist in a deaf world. No one will pay attention. Well, Beethoven was deaf when he wrote his Ninth Symphony, but there aren’t many Beethovens in the world. It’s extraordinary that people really, even though they’re convinced, they have a great deal of difficulty working in the environment where their managers are pushing them to do other stuff, everybody’s got to get into test, quality is not a priority, their peers are racing into test, everybody’s getting credit for getting into test early but they are not recognized for doing quality work.

As I said, I talked about this engineer at Siemens, in Siemens research. He did beautiful work, but no one knew. He was doing it all by himself. He was his own audience. No one else was paying any attention. So beautiful work in software does not get noticed. It’s a tragedy. There was one guy I ran into in IBM Kingston-- exactly the same thing. I remember it was a long story. Remember we talked about-- one of the things we’d put in place at IBM before I retired there, from the quality job, we discovered that over 20 percent of IBM’s APAR(Applied Programming Assistance Request) repairs-- these were customer-reported defect repairs-- over 20 percent of them actually injected defects.

So we would put out fixes that were themselves defective, and the fixing process was in the maintenance organization; it wasn’t done by the developers. So this became a big issue because when we really looked at the data we discovered it was enormously expensive – like 40% of our programming service costs. I mean, fixes get slapped in. People don’t install them slowly. So when you have fix problems, you’ve got crises. And we had lots of them. So we started something called APAR certification, where before the maintenance people could ship a fix, they had to verify it with the developer. So we put that in place, and we had this procedure everybody was supposed to follow.

I remember, I was up in Kingston, and I was doing a random check, just talking to experienced engineers who were doing development. I called in one engineer they sent in to get a random check to just see. And I used to like to do this, go talk to working engineers, see what they're doing, what is going on. So I asked him about APAR certification, what he thought about it and how it worked. And he said, "What's that?" I was really irate. I said, "What do you mean? We've got this enormous effort getting APAR certification across the company. Here's why we're doing it," and that sort of thing. And I said, "How come you don't know about it?" He said, "Well, I've got 30 modules in the core of the MVS system that I'm responsible for, and I've never had a defect in any of them." And I said, "You win. You don't need APAR certification." Well, no one knew that. We've got this guy doing extraordinary work and no one knew it. But that's what our world is like. You have these people doing extraordinary work and no one recognizes it. It's terribly sad.

So in any event, I discovered the PSP-- the personal software process-- much as it worked, and much as it enabled people to do truly gorgeous work, very few people really have the personal discipline and the drive and the commitment, and they certainly don't have the environment to permit them to do that work. So that's why I concluded that we really had to move from the PSP to the team process. I had to put together a team environment where you could really do beautiful work. And that's what led us to the TSP. Okay?

Booch: Got it. Very good. Let me level-set here. What year would this have been you're starting to see the transition to the team software process?

Humphrey: It was 1995 I published the PSP book-- the end of '94 actually, January '95. We spent '95 trying to put all this stuff together, running into all these experiences, and it was in early '96 it became pretty obvious we had to put together a team process. So I started to construct a TSP, team process. Could we take a break for a few minutes now and then.

Booch: Delighted to. I need to take a bio break as well and grab something to drink. Should we give ourselves five minutes or so?

Humphrey: That'd be great.

Booch: Very well.

<break>

Booch: And we're back live again. So let's pick it up where we were.

Early TSP Trials

Humphrey: As I said, we'd gotten the PSP in place. It became clear that people really weren't able to use it. So I had to build something that ordinary folk-- not zealots like me-- could actually use on the job. So that I decided was a TSP-- a team software process-- and I wanted to put something together. So starting in about January '96-- remember I talked about our process development process? I pulled out my process development process, and I started to develop a TSP process. I could go back and get the exact dates; I've got all the records of that stuff.

But in any event, I did, and I got it together during the spring and summer, and I was trying to figure out where to use it, and the people at Embry-Riddle said, “Hey, let’s us do it.” So what they wanted to do was to use it for I think a team of four or five of their graduate students who were going to do a team project, and they did team projects. They’d all been PSP trained, now. So they were trained, they were qualified. So that’s what we agreed to do.

So I put together the process and had it all defined and ready to go, but unfortunately, in September, when the project was to start, I had to be in Australia to give a keynote at a conference. So I just packaged up the process and sent it to them at Embry-Riddle. So they basically went through it. They followed-- our processes by the way have things called scripts. It’s what I call an operational process.

So the process isn’t a big document thing that has paragraphs. It has steps in a script which you follow. I’ve got copies or versions of them in my textbooks and stuff. But it’s got a series of steps, and if you read what it says in the steps-- let me pull up a couple here-- they’re very direct. They just tell you what to do, step by step. So it’s not complicated at all, but it does have specific-- let me pull up a process. “Schedule planning template: Enter the following.” I mean, it starts off with a purpose. “The purpose is to record the estimate and actual hours expended by a calendar period, and relate the task plan value of the calendar schedule, plan value earned, value stuff, calculated adjusted planned and earned value <inaudible>.” And then, “General: Expand this template or use multiple pages as needed. Complete it in conjunction with the task planning template.” Then I go, “Enter your name, date,” etcetera, “Week number, when the project started. Enter a week number, typically starting with one. For very small projects, it may be more convenient to use days instead of weeks.” Then the next one.

So basically it goes through-- this is the instructions for the template, and then the actual process script starts with entry criteria, and then planning. “Step one: Produce or obtain a requirement statement. Use the probe method to estimate total new and changed lines of code, required time and the prediction interval. Complete the size estimate template. Use the probe method to estimate the development time required.” So it goes through steps, very straightforward, and we have scripts like that for all the steps. The reason I had to do that is kind of interesting.

And think about it this way: when you tell somebody to produce a plan for a program, most programmers won’t have the vaguest idea what you’re talking about. So they’ve got to go off and sit down and figure out, “How do I make a plan?” They don’t know. So they have got to kind of figure it out. Even though it’s not very complicated, they don’t know; they don’t have a process; they don’t know the steps. So they’ve got to figure out step by step what to do, and then they do it. So that’s an enormous waste of time. It’s fairly obvious. We know how to do it, so we just give them a script and say, “Here, just do this.” So that’s what we’ve done, and it works extremely well. So that’s what I sent to the folks at Embry-Riddle. How do you do a TSP project, and that’s what I had done. When I developed the process, I developed all the forms, the form instructions and the scripts for every step of what they had to do.

It took me several hundred hours of work. I mean, it’s a lot of work, and it is non-trivial… There’s a big, big pile of stuff. So I sent it over to them. When I got back from Australia, I went over to meet with the team, and they’d been working on the job about three or four weeks. It was a good team, and I had this session with them, and they were kind of laughing, because what had happened was they had kicked it off, following the process, and as they were going along, they were getting kind of frustrated because they had to move faster. They were in a hurry. So they kind of skipped the process and just started doing what they had to do anyway. They just did the things they knew how to do like design, code, and test.

And this is sort of what you'd expect programmers to do. But they got a little bit along like that and they realized they were totally lost. They had no idea where they were. And that, by the way, is where most programmers are most of the time. They don't know where they are on the project and what to do next, and that sort of thing. So they basically said, "Okay, we better go back." And so they did. They basically said, "Okay, we are going to follow the process." So they did, and they did a marvelous job. They went through and they produced a system. Unfortunately, it never got put in production because one thing or another happened over there. It was a system to manage the flight line at Embry-Riddle. They were going to develop a flight and instructor scheduling system. And they put together the requirements. They went through the design. They did the whole thing. But the process basically gave them an anchor, so they knew where they were, what they should do next.

So it was quite helpful, but they had a problem getting started; their plan wasn't very good; lots of stuff. It was pretty obvious from that that they needed guidance when they were starting the project. We couldn't just send it to them and say, "Go start." We needed to have an initial-- we decided to call it a "launch" of a project. And we needed to have somebody to coach the project. It occurred to me-- I talked about my wrestling coach-- really high performance teams need coaching. So we decided we were going to have TSP coaches. We originally thought of them as coaches for the launch, so they would basically launch the project, and then the team would just go do it. And so we did. So we went and did a couple of industry teams. I think we did a couple up at Harris, up in upstate New York. Of course, they got reorganized and everything. So we got started, and then everything got canned. But we had a couple of project launches and some early experience there, although they never really finished the project.

I kept running into this, by the way, in organizations. You get halfway through and they get a re-organization, and you get a new manager. The continuity in business is so appalling that you literally can't keep stuff going. In most of the places where we worked, a fairly high percentage, you get something started and then the sponsor moves on and something else happens, and it's dead. It's gone. You've wasted an enormous amount of effort. And it happens all the time. It's just astounding. And it can't be just us. It's got to happen on everything going on in industry. So it's kind of frustrating. I don't know if it's just the U.S. or not, but the dynamics of the management system we've got and the ability to have any continuity for improvement in business is awful. I certainly run into that. It's a real problem with the software community.

The Teradyne Team Launch

But in any event, so I started, and I was the launch coach for a number of projects that we launched. I remember doing one at Teradyne out in Chicago, and that was a marvelous project. It went extremely well. Bob Musson was the lead process guy there, and he brought us out. So we did a launch. I launched a team, and had nine software people and five hardware engineers. This was a very early team, and we had no problem using hardware engineers on it. They were designing a chip and a special processor-- or they weren't actually designing the chip; they were designing a little special purpose machine it was to be a line tester for telephone networks.

And they were basically-- they were trying to find line failures and it was for the telcos in Europe and it was the new version of the program that they had or the product that the company had previously been selling over there. And-- but this was to replace that and it was supposed to have some new technology and AI logic for analyzing line failures and all that sort of stuff. So it was pretty aggressive program. And I was there and I remember we started the launch with a meeting with the management team. And this

turned out to be a fascinating experience because most of the teams that we've worked with have never had a meeting with management when management says what they want the team to do. We start the launch that way.

So the opening meeting is with senior management, we bring in marketing people who will describe what the customer wants, the customer situation. And the reason for this is interesting and that is that a lot of the decisions you make when you're producing a system are tradeoffs. And they're tradeoffs of what you can do technically and what the business can afford and what the customer wants. And so the people who are really making those tradeoffs in very many cases are the programmers. Nobody else is involved in enough of the detail to understand it. So the programmers are making these very sophisticated business tradeoffs without the vaguest idea of what's going on in the business.

And so we start the launch by bringing in management to give the team the business perspective and bringing in the marketing or customer people. And we bring in customer people if we can, who will tell them what they want and why. And then the programmers have a perspective for making decisions. And what's interesting is during the launch we open the launch with management and then we basically close the door and have a meeting with the team. And I'll come back to that one on a later team launch we had with Boeing which was fascinating.

But in any event we close the door and the team goes off and puts together the plan. So we went through the opening meeting with this team at Teradyne and the management was absolutely firm it had to have the product in nine months. And I poked at that a little bit. In the opening meeting the team is supposed to ask questions but they practically never do. The team leader may ask a couple but by and large management says what they want and nobody says “boo.” So I asked a couple of sort of quick questions, I didn't want to say too much because you don't want to tell management that they're crazy and that sort of thing – at least not until you have a plan.

They're not—the team doesn't know enough to argue yet. So basically the team is in listening mode and the management was really very firm. And so when the meeting broke I went out to the washroom and then I joined them in the meeting room and they were in turmoil. They were irate. You have heard of the storming phase? Well they were storming. And so, you know, they were saying, "This is impossible we can't possibly do it in nine months; this is crazy the last project took two years and it was a disaster and this is more complicated than that."

And so I asked them I said, "Whose date is the nine months?" They said, "It's their date, management’s date." I said, "Okay. So what do you want to do?" And they said, "We ought to go back and tell them it's crazy." I said, "If you do what will happen?" They said, "Well they'll beat us up and we'll, you know." they kind of mumbled around and they finally concluded, "Yeah we'll end up that we will try if you insist but it's a very tight date we don't think that we can make it."

I said, "Okay if you do that now who owns the nine months?" They said, "Oh we do." And I said, "Do you want to own the nine months?" And they said, "No." I said, "Okay. So here's what you can do". I said, "You've got to make a plan and do your utmost to make a plan that will end in nine months. If you can't do that then you'll know why and you'll have various alternatives you can say with some evidence that you can do the following. But at least you'll be able to understand in some detail exactly what it'll take to do the job and you'll have a foundation for debating the subject with your management." So they bought it. So we went and made a plan.

And the launch process goes through starting there, the second meeting we actually go through-- I mean basically we later added stuff to the launch, but we didn't do all of this at the Teradyne because we kept learning as we went ahead. But the process fundamentally is, we come out of the management meeting, you go into a meeting on basically goals and roles. What is it we're trying to do and how are we going to divvy up the work among team members? Now this wasn't divvying up the project work it's divvying up the management of the job.

And so the TSP actually has role managers and so we have a planning manager and a design manager and a quality manager and a process manger and a test manager et cetera. And these are people who have specific responsibilities for aspects of the job. The test manager for instance doesn't run all the testing. But the test manager is the person… if there is any issue with testing you give it to the test manager and he or she willl take care of it. Make sure the test tools are ready in time. That they're thinking about testing when they're doing requirements work, et cetera. Same on design: what methods are we going to use? Make sure everyone is using the same design approach and deciding what is the notation we are going to use and what are our standards? The quality manager is looking at the quality data with the process manager and deciding how to do inspections and this sort of thing. So these are the jobs these role managers do. And what's interesting is that on most projects today, no one does them. They're all left for the project leader to handle. And the project leaders are handling all these mechanics. They're trying to track the plan and this other stuff and they don't do it very well. They don't have time.

And so fundamentally most of the stuff that these role managers do doesn't get done on a typical project. And so what-- the basic rule we use on the TSP is that team leader's job is to make sure every definable task is given to somebody on the team. The team leader's job is to do all the indefinable tasks. He or she handles the motivation, the tracking, protecting the team from management—staffing is a constant problem -- the minute you launch a team, management wants to borrow somebody. And you've seen that before.

Booch: Oh yes.

Humphrey: And so the team leader's job is to protect the team to make sure issues get handled, to work on risks, and I mean a whole bunch of stuff and motivation and measurement, evaluation, process discipline, reporting. Team leaders are very busy people and they are the ones who actually make the difference between whether the project succeeds or not, the team leader and the coach. What's interesting you think about like a ball team that is in the cellar, who do you get rid of to fix the team? You don't get rid of the players you replace the coach or the manager. And we don't normally have coaches in the software community. Coaches are extraordinarily important.

The difference-- and this is one of the issues I keep running into what's the difference between a team leader and a coach? Why can't the team leaders be coaches? Well we're probably going to have to come to more manager-coaches just because coaches get laid off and team leaders don't. We're running into that now in the in the economic squeeze. Unfortunately, management doesn't recognize the enormous value of coaches.

Booch: This reminds me of the story that you told me earlier in your wrestling career how the same person you and under the one Olympics quality coach you guys performed very differently than with the subsequent coach.

Humphrey: Exactly. Exactly. And we don't see that in the software… We don't see it with development teams at all. And I've not so far talked about my experience with my early teams in engineering. When I was managing development teams I was really acting more like a coach. If you remember, I was asking people what they were doing and focusing on why you're doing that and this sort of thing. I wasn't really saying do this, do that, do that. We had a lot of things that we had to do and we had to put together a plan but I was amazed at how motivating that was. People love it and it worked like a dream. And it worked extremely well here in Teradyne.

So anyway to continue with the Teradyne team. We went through-- first they went through goals and roles and set that up and then we get into a strategy session where we figure out what's the process we're going to use. What's the strategy-- well we do the strategy first. What's the strategy for developing this product? And people aren't sure what the strategy is but there is an enormous range of strategies you can have as you well know. And that is you can build the whole thing in one big bang, you can decide to build versions of it. You can use all kinds of cyclic processes, you can prototype stuff. There are lots of different ways to build a system. And you may not make every part following the same strategy; you may have several strategies depending on what are the tough problems You want to identify the real nuts that will have complex technical issues and get them on the table early. And maybe you want to prototype them. All kinds of stuff. And so they work out the strategy first. You have quite a bit of discussion on that. What is it? How are you going to do this and why? And you think about the risks and what are the problems and exposures you've got.

So we think through alternatives, you may want to just build it the way you did the last one. And then once they lay out a strategy then we say, "Okay well now what's the process you're going to use?" And notice we have all these unit processes that they've got. They know how to do configuration management work, how you write modulus and test them and all that sort of stuff. But then you have to weave that together into an overall process that you can use for actually making a plan. And so we have the team do that before they make a plan. And the purpose of this is to make sure you don't confuse people, you don't confuse how to do the job with how long it'll take. Because when you start right away to make a plan and you're thinking about how you're going to finish it in time while you're planning. It's amazing how often people came up and say we don't have time to do that.

And the contention that we follow with the TSP is that you want to do it the right way and the issue is to figure out what the right way is and lay out your plan based on that. And what you'll discover is if you're really doing it the right way it is the fastest, and cheapest and highest quality way to do it. And so we have the team start by figuring out what's the right way to do it and only then do they estimate how long it'll take and the resources. And that turns out to be a very powerful way to do it because now the team has something to defend that they believe it.

And so the team has a big debate about the process flow and how do you want to do that? Do you really want to do this do we want to inspect every module? And the guys, you know, once they've got data and quality information they realize: yes, we're going to inspect every module and yes we're going to do this and we're going to do that, et cetera. And you get the teams to buy that they're going to have personal reviews and that sort of thing. So they went through all of that and I talked about goals earlier, the goals they put together; management goals and team goals.

And in the second meeting the goals they actually set goals for the yields (% of defects found) they want to find in their reviews and what they're going to accomplish in terms of various things. And so - there are

a whole series of goals that they've already made when they're going through a lot of this planning and so when they finish that they finally have got a process defined, they've got a list of all the products they've got the build. They put all that together and then they make a team plan. And so in the next meeting, the whole team works through how big are the parts how long will they take to do each step and they lay that out. They're not talking about who's going to do any of the steps yet. They're just laying out the whole job and how much time it'll take and what the effort is and they also lay out how much time the individuals are going to have.

The Task-Time Measure

So all of a sudden you have kind of an idealized plan that says if everybody was busy full time and could do any of the jobs, here are the total jobs that we've got and here are the times that people have got available and that works extremely well. Now one of the things that we found right away is that we're not talking about calendar time. Remember now I mention in the PSP that we measure the time and the size and the defects. And the time we measure is the time for tasks or process steps or phases. And so that's task time, it's not calendar time and so the question now is, what's the relationship between task time and calendar time? Let me just ask you that. Suppose you have a project to do and you list all the tasks for that project and then you put against that how much time you're going to work and then you work and you do the job. At the end of the day or end of the week, what percentage of your time was task time and what was something else?

Booch: My guess is that you're task time, if you're a code warrior, is probably something like 20 to 30 percent at best of your calendar time.

Humphrey: You're probably right.

Booch: There's this study I read by a sociologist from Berlin of all things, who did a study of how developers actually spend their time and he, through the study, enumerated a long list of things that cause friction and he was amazed himself how little time the people actually had to do things because they were busy with meetings and things didn’t work and stuff like that.

Humphrey: Well the thing we found was that the task time in various organizations today without the PSP and TSP, the percentage of time-- or in an 40 hour week an individual engineer will spend somewhere between three, five, ten maybe 12 hours a week doing project tasks. The rest is all kinds of other stuff. And people are very surprised at that. And what we find with the TSP is astounding because now instead of just sort of hoping you can get more time in, the engineers are actually measuring and managing their task time.

And they're tracking it every week. They look at it, they've got a tool, they can follow it. So we find that a team will start-- they begin at eight to ten task hours a week and after a period of weeks, it takes a while, they'll get up to 15 to 16 to 18. We' have teams running over 20 but not many. Most of them are running in the 15 to 18 range. And what does that do to productivity? I mean that's enormous. You double the task hours per week for the individual engineers and you discover that you have in fact doubled productivity. It's amazing and it's an extraordinary result. We actually did it-- go ahead.

Booch: No I was trying to level set where we are in the time frame where the TSP really reaches this level of maturity. Because we're now talking around the turn of the millennium now because-- would that be correct when your data gathering and such and your experience of the projects have reached this kind of fruition?

Humphrey: We were in about 1997 with Teradyne.

Booch: <inaudible>

Humphrey: And we basically come to most of these conclusions before we did the Teradyne launch. We learned a lot from the first couple of projects, it's amazing when you've got all this data and stuff and you look at it. You learn stuff real quick. So we, as I think I mentioned before I had a period of years in here where I learned more in the brief period than I ever really learned in my life. But it's an unbelievable education.

So in any event, yeah the task time thing and what we did was evolving the process with the task management and all of that. We gradually evolved it but that was there at the base starting at the very beginning. And so we tracked that and so I went through with the team, they completed their overall plan, determined the available hours the people had-- remember now when the people lay in their time we have them actually look at the calendar. When are their holidays? When are you taking vacations? What's going on in your life?

So we actually had each of the engineers lay out his or her personal schedule for the whole period of the project. How many hours they were going to be able to have during Christmas breaks and Thanksgiving breaks and you name it and so they laid that out and then we spread it over the calendar and the calendar was 18 months, not nine months. And so the team was kind of shaken by that.

My God they couldn't believe it. “What are we going to do about that?” So we had to make another step now because this is still an idealized calendar. And I said, "You've got to make a real schedule." And so we had the overall plan before we went to make the real schedule we made a quality plan. Remember these are all PSP-trained engineers they knew how many defects they injected and found themselves, and what their yields had been in PSP training.

So making a quality plan was no big deal. They basically estimated how many defects would be injected and removed in each phase and how many would be left at each step and how many they'd find in test and how many would be found in the field and all that sort of stuff. And they had it all there and it was great, marvelous.

And so they put the quality plan together and then they put together after the quality plan they went into we call it Meeting Six where they put together the individual team plans. So they-- now you take the plan that they've got and they start to allocate tasks to engineers. And so they go through that and they go through the allocation and the engineers each produce a personal plan which is rolled up into a team plan. So now every engineer has a plan for what he or she is going to do to go through this project. But when they make the detailed plan they don't make it for the whole project. They make the overall plan for the whole project then they make the detail plan for the next two, three, four months. And they don't go very far out because you can't. When you actually follow the plan lots of stuff will change and it's a waste

of time to make detail plans beyond more than a few months. So we make it for the next few months only and they lay it out and then we do load balancing.

And you'll discover that when you put a team together there's the lead designer who has got, you know, hundreds of hours of work to do and you'll have somebody here who has just arrived who has a few dozen hours of work to do and so the team really has to work together to do load balancing and figure out how do we move this work around and unload the lead designers and the hot shots and get everybody else to work. And so we have to team people up and do all kinds of stuff. And this is a big team negotiation. Who can work with who and what can they do? And that's part of Meeting Six and we go through load balancing and get that done. And that's enormously effective. I mean you really do end up with people thinking through how they're going to work together. And that doesn't happen on most teams. But it's done and they have this plan. They know how they're going to do it. And then after Meeting Six they've now got a plan they got a schedule they know who's going to do what and they're ready to go to work on Monday morning.

And then they go into Meeting Seven where they make a risk analysis. They go through what are the risks on this thing and they go down the list and what are the problems and whose going to handle each risk? The team members pick up as owners and a lot of them the team leader owns, but there are others and we-- they identify-- they rate the risk as high, medium and low likelihood and high, medium and low impact. <phone rings> And the high high <phone ringing> the high high that sort of thing they put together mitigation plans for them.

So the team now has a big list of risks and they have the top priority risks and they have mitigation plans for them and then we go into Meeting Eight and the team now puts together the management presentation. And they've got an enormous amount of material. They know exactly what they're going to do. And it's a very impressive result. These guys have got a complete plan and it's really an amazing result. And they're committed to it, they believe it. This is a plan that they are committed to, they know what it is, they know how to go about it and so we went back to the management meeting. Now remember management said, "We've got to have it in nine months, there's no alternative." And the team had a plan for 18 months. And so there was a lot of nervousness when we started the morning meeting it was Friday morning. And the team was all there and the general manager came in and the marketing VP and that sort of thing.

Presenting the Plan to Teradyne Management

And so the team started. I introduced them briefly and then I turned it over to the team leader who went through the presentation of what they did, how they made the plan and that sort of thing. And then they started to go through the plan. And when he hit the 18 months everything stopped. And the general manager started poking at him and asking all kinds of questions and he was beginning to soften up, you know, all right it sounds like you're right and the marketing VP blew up. He said, "You're going to kill the company. We can't possibly wait 18 months."

He said, "The competitor is delivering a better product right now." And everybody was sort of stopped and stunned. How can we live for 18 months when we have a competitor that's got a better product out there today? So I asked him I said, "The competitor actually has a working product in the field right now?" And he said, "Yes." I said, "When did you think they started developing that product?" He said, "I don't know." I said, "Probably a year or two ago right?" And he said, "Well yeah probably." I said, "Why didn't you start

then?" He said, "What do you mean?" I said, "Your job is to anticipate the market. These guys can't fix that problem for you." And he kind of mumbled and sat down.

I really kind of beat him up there. But the general manger then bought the plan. We came out of the meeting and the team turned to me and said, "Watts, we could never have done it without you." And I said, "The hell you couldn't. You certainly could have." I said, "You've got the story." Well it turned out the next week the Marketing VP came back with a bunch of hot shots to the team and said, "Okay show me. What the hell is going to take so long to build this thing." And so they did and they pulled out the plan and they went through it and they spent hours and finally at the end this guy had really listened and he had some good guys with him and he kind of scratched his head and said, "You've got a lot to do, don't you?" And he bought it. And it turned out they ended up-- the marketing guys were so impressed, they brought in customers in to see the architecture that the team had come up with and the design they had and everything else and they didn't lose a single customer. They kept them all. And what's interesting is that we have had that experience with every team. I have yet to have a team lose the management debate. <phone rings>. The teams are really winning and it's exciting <phone rings> <inaudible> <laughs>. We can keep talking through it if you don't mind.

Booch: That's fine keep talking through it, the transcriber will just go nuts here for a little bit but I'm sure they can do it.

Humphrey: But in any event, so the teams win these arguments they've gotten better and better in terms of bringing in alternative plans and this sort of thing they'll come in and say, "Look with this number of people it will take us this long." The first Microsoft team for instance, they came in with ten alternate plans. They said to management, "To meet your date we'll need two more people and we've got to have them on this date and they've got to be PSP-trained. With the team we've got it will take this long or if we reduce this function we can do this."

And so we guide them on having multiple plans to go in and give management various alternatives how they want to do the job. We also typically have the coach or the team leader go in and meet with the senior manager before the final management meeting so that we don't get him surprised. But we find the whole thing only works if we've had management training ahead of time. So we have an executive-- we call them an executive seminar where we put senior management through what this is all about and how it works and why. So you get them to understand the dynamics of what this process is all about. And then we have training for the team leaders and lower level managers where they go through how do you manage projects like this?

The executive seminar is one day, very straightforward, no big deal and we will put typically senior executives really top executives will go through that. We recommend that it's got to be top executives. Without senior enough management, the process won't stick and that's the problem that we've had at Microsoft and several other places. The top person is the one that keeps it going. And so that's crucial. And then the managers we like to put them through a four day course. If the managers have been through the exec seminars it's only three days maybe I'm wrong maybe its two days without the seminar. I've sort of forgotten, but I've taught it but I've sort of lost it at the moment. But at any event it's only a couple of days. And they learn it, they go through what it's all about. How do you manage teams? What's the launch process? How do you use the data and what do you do?

And then we don't train the engineers until their managers are trained. And during this whole cycle typically in the executive seminar, the management team picks the first teams that they want to use the TSP and they identify the chain of management and that sort of thing and then they go ahead and train all of the managers in the management chain between the top executives and the team and they train-- and then we go to training the team members.

Our original PSP training took two weeks. Everybody spent two weeks, but managers typically just don't want to have their developers take two weeks for training. They had to shut down for two weeks to get their teams trained. And so we had actually gotten-- we found surprisingly that as people get more and more receptive to this--initially people objected they couldn't believe that the PSP made any sense. By and large we're now running into lots of places where they're quite receptive. "The PSP, oh yeah we heard about that, how that worked." And they're not fighting it anymore. And so we find that we can get the basics across in a week, enough to actually go through with the team launch.

Booch: So the initial times it was you doing the training can you tell me the growth of how you developed in SEI teams of people to go off and do this training? But was it in fact you primarily in the first and tell me how it grew.

Humphrey: Well I did it at the very beginning. Remember there was some SEI people in the first PSP course that I taught that went into this.

Booch: Yes.

Humphrey: One of the key guys there was Jim Over. Ever heard of Jim?

Booch: I have not.

Humphrey: Okay. Well Jim Over is the leader of the TSP group at the SEI. I work for Jim. I work with Jim. He laughs when I say he's my manager. But as you know I really don't work for anybody.

Booch: Okay.

The TSP Team

Humphrey: We formed the TSP group in about 1996 or '97 and Jim Over became the leader of the very small group. Jim Over and Dan Burton joined us. Dan was an interesting guy, by the way. Dan had been a Major, I believe, in the Air Force and was in the program office that formed the SEI. And so when he retired from the service he couldn't go work at the SEI right away so he worked for a year in one of the local software companies. He was doing software development, and then he came and joined us. And so he's been on our team ever since and been a marvelous asset. And we had a bunch of other people. Julia Mullaney was in my first PSP course and she joined us and then we gradually built a team. Rather slowly, we didn't have much funding and we've been fighting the DOD on this thing since the beginning by the way.

And I basically never got funding for this stuff. It was my own work and we basically had to almost self-fund. We got some funding and some backing but not a whole lot and so we have been scrambling to try to get this thing done ever since. And it was years of that with no one really understanding what it was or buying it or anything. Even in the SEI. And the DOD people who have worked on this for some reason they had a very negative view of the PSP and the TSP and I think it was because they had a negative view of me and I'm not quite sure why. But they fundamentally got working on the CMMI integrated thing and I was focusing on software and I guess they felt that I was too narrow or something. I don't know.

Booch: And if my history serves me correctly this is around the time of the DOD having it's specifications for software development that was-- what was it 493 or 2167? I'm trying to remember the numbers.

Humphrey: Well there was 2167 in there.

Booch: That's right. That's right and, you know, as I looked at that relative to the work that you were doing you can see that these were on very different paths. And your work hadn't yet influenced that activity yet.

Humphrey: Exactly. Exactly. And it really hasn't yet. I'm not sure who we're influencing except ourselves and industry. Not influencing that crowd. But in any event, so we realized rather quickly we had to have people teach this stuff. And so fundamentally the group watched me teach and they started teaching it. And then we realized rather quickly we needed more instructors because we were starting to run into more companies. So we put together a course to train instructors and we put together a course to train coaches. And then we started working with other people who wanted to form their own companies and go out and introduce the PSP and TSP.

And that's sort of what's been happening over time, although typically the bulk of it is still us. Although we find that the people who are trying to make a business out of this are not doing all that well yet. But we do when we start with a company like Microsoft or Oracle or Intuit or people like that, Adobe is another one, who buy it and they're doing it and they decide-- we train their own people to be their own instructors and coaches. So they become self-sufficient. And so that's what we do there so that people can do it on their own and we can begin to grow it and expand it. And they can train their own resident coaches. People have to have coaches on site because having remote coaches come in is difficult. We have to do it when we start but we really like people to get their own coaches.

Booch: So that brings us to the late '90s then. And it sounds like, you know, the next several years were still largely consumed with growing that activity. And growing the teams and spending a lot of time out with real customers doing this. Is that a fair characterization?

Humphrey: Yeah. We had to improve our courses and the training and the tools and begin to build credibility and get more people familiar with what we're doing and what's going on. And that worked fine. And so we've made a lot of progress and there are a lot of people using this. We've got hundreds of teams that now use-- or companies that now use the TSP.

Booch: Now.

Humphrey: That's enough on how we started the TSP.

Booch: It's my recollection that you and I ran into other around that same time as well. Wasn't it on, gosh it was on-- it was some Army project that you and I were together on. Was that?

Humphrey: That was early. That was very early, that was in 1989 I believe, '88 or '89.

Booch: How quickly I forget.

Humphrey: That was the AFATADS project and it was at Magnavox.

Booch: Yes I remember now. Yes. Yes. Yes.

Humphrey: Out in the middle-- Indiana I think someplace like that.

Booch: Yes. Yes. Yes.

Humphrey: You and I were there and I remember the people going by the door that's Grady Booch. …your Ada book…

Booch: That's right.

Humphrey: And so, yeah. No I remember that well. We had actually-- this was an audit as I recall for the Army.

Booch: Yes. Yes.

Humphrey: And as I recall, prior to the audit, I had led a CMM assessment of that facility. So I'd already done an assessment of the location, I knew a lot of what was going on inside the group but I promised them that if I was on and I-- this was part of what we agreed with their management because they were-- we were called in by the Army to do an audit of them and I was reluctant to participate unless it was an effort that Magnavox was actually participating in. So there would be a joint effort and we report to their management and they finally agreed with that so that we had that worked out before I did the assessment there.

Booch: Right I remember that now.

Humphrey: Yeah. So this was way before the PSP and TSP started.

Booch: How quickly I forget. Well I hope I have as good as a memory as you do in the order of things.

Humphrey: I plug them in to where I was to this whole thing. I sort of see it in terms of the time frame but when I published the book on that and so I sort of have a time scale there. Well, as I say the cycle we went through now was to expand this and there are a few stories that I want to tell you with what we're doing with Mexico and some other countries. What's happening, the large system issues, what we need to do there. What's ahead, what we're doing as we move out of just software into other fields. There's

some things there that I think are interesting and then I want to get into sort of what next and where we're going.

Booch: Very good. I'll tell you what let me stop the recording here. And then we can plot what those next steps are. Let me stop now. There we go.

Day 4 AM, June 22, 2009

Booch:.Back here again on, this is day four, is it not, Watts, of our recording? Or day five, something like that? It is here a bright Monday, June 22 [2009]. Grady Booch again back with Watts Humphrey continuing our wonderful interview of this and that and everything in between. In fact, the last thing we discussed, Watts, was a little bit of the origins-- well, not a little bit, a lot a bit of the origins of PSP, how it came to be, its beginnings, but now let's turn the topic of discussion to where is it now and kind of where is it headed? Can you take us from there?

Humphrey: Great. Love to. And, by the way, I very much appreciate all the time you're taking. We've had some 14 hours of interview so far so it's sort of babbling on and I must admit it's great fun to relive all this and talk through it.

Booch: <inaudible> eyewitness to it.

Humphrey: Great. Wonderful. Okay, well, where we stand right now, very briefly, we've got a group of about 10 people at the SEI in Pittsburgh, a couple others work remotely. I work, for instance, here in Sarasota, Florida. Jim Over leads the TSP team and I work on the team. I'm a member of the team, although I originated all of this stuff, I'm not there and I really am not able to take all the trips and go around and do all of those things. I decided long ago that my main focus would be on sort of the longer-term stuff, putting together textbooks to support it, sort of ideas, going out and giving keynote speeches, that sort of thing. So I'm sort of the grey-haired eminence behind the scenes here going out and talking to lots of people and trying to be visible and, unfortunately, my current cancer is keeping my travels limited and it may for awhile. But, in any case.

Six Months to Live

Booch: <inaudible> would you be willing to sort of interject here about your current cancer? You and I have talked about it. We don't really have anything on tape there. Would you feel comfortable spending a few minutes talking about that situation?

Humphrey: Of course. Sure. Yeah. Well, I remember exactly when it hit. It was February 19th this year. I was feeling great. Came back from lunch with some wonderful friends and, about 4:00 in the afternoon, I got chills and fever and I just couldn't get warm. I bundled completely up, went to bed. About 9:00, I got up and called Barbara about, you know, “I don't feel very good” and I passed out cold. <laughs> So she called 911, they took me into the hospital. This was late on a Thursday night, about 9:30/10:00 Thursday night. They got me into the emergency room and started doing tests, obviously, to see what the heck it was. They got me in a room there and, during the day on, I think, Saturday, they came back and said I appeared to have a blood infection. It's not easy to get blood infections. But, in any event, I had an infection in my blood and I asked the doctor, "How serious is that?" and he said, "If you weren't in good shape, you'd be dead." So I.

[audio off then on]

Humphrey: Now, that got my attention. Now the challenge was to find out what caused this infection and, of course, they started me on all kinds of antibiotics and they really blasted me with them. Are you still there?

Booch: Yes, I am still here.

Humphrey: Your picture disappeared for a moment. Okay?

Booch: OK.

Humphrey: So they really blasted me with antibiotics and Sunday they took me down to do various scans and things and Monday the doctor went down my throat with one of these things to probe and see where the infection or where the problem was and they obviously produced a report from the CAT scan and all the other stuff that was going on. My doctor faxed it to my daughter, who's an M.D, and Tuesday she called me and said, "Dad, you've got cancer of the liver and they say it's inoperable."

I said, "That doesn't sound good." Well, the doctors hadn't told me and, of course, they had more tests to do and stuff but my daughter knew I wanted to know what was going on so she told me. So I arranged to get out of the hospital right then. They'd finished all the stuff, antibiotics and stuff and they finally let me go Wednesday morning and I went to see the oncologist and he said, yeah, it was cholangial carcinoma of the liver, that those cancers do not respond to primary treatment with either radiation or chemotherapy and that, in my case, mine happened to be inoperable.

I said, "Well, how long have I got?" and he said, "Three to six months." This was four months ago. So, you know, that's a bit of a problem. <laughs> My daughter had said that the hospital that had the most publications on this kind of cancer-- the doctor, by the way, told me what we should do is to look for somebody who had an experimental treatment or something that may deal with it. Other than that, I'd just have to wait and we could decide whether I wanted to do chemo or radiation or not but there was no evidence it would help extend my lifespan.

So, in any event, it was kind of dismal. So my MD daughter came up with Sloan-Kettering in New York as the hospital that had the most publications on this and seemed to be doing the most. The oncologist contacted them and they said, "We can't even talk to him for three weeks," which didn't sound too good. <laughs> Not what I would call responsive. Another daughter actually had a college classmate who [tape glitch] she thought had gone into cancer research. This is a long story but I might as well get it on the table. And so she sent him an e-mail, got his e-mail address off the Internet. His name was Dr. David Fisher. Turned out she got the wrong Dr. David Fisher's e-mail address but, by sheer staggering luck, he knew the right Dr. David Fisher and forwarded it to him. He's at Massachusetts General Hospital and he had a very close friend who's the top oncologist for liver cancer, GI cancers at Mass General. He sent him the e-mail and he got hold of my daughter, like, within the hour and said, "I'll see him Monday."

Well, that sounded pretty damn good to me so we headed up to Boston and, when we came in Monday, he'd gotten the top cancer surgeon from Mass General and the director of all the radiation therapy and stuff, both of them to see me Monday afternoon, squeezing all this into their very busy schedules. I mean, it isn't like these guys have nothing else to do. So I was enormously impressed. They were thinking of me as a person, not just another case. The surgeon said, "Yes, there is no way we can operate on this

cancer. It is, in fact, impossible to remove it to surgery." Then the director of radiation therapy there from Mass General said, "It turns out, we have an experimental program with National Institutes of Health using something called proton radiation."

It turns out proton radiation-- I had the guy describe what was going on and what they do is they use a cyclotron and they actually radiate the cancer with protons and the proton radiation has a particularly powerful profile when it actually impinges on the body because, if you use high-energy protons from a cyclotron, the protons go right through the outer layers of the skin and stuff and then they gradually slow down and finally get captured at some depth, depending on how much stuff they've gone through and they dump most of their energy where they're stopped. About 70% of the proton energy goes right to the cancer.

They have a way this guy had come up with to do that and the real invention was that the liver moves when you breathe. So normal radiation treatment doesn't work with the liver so he'd come up with a way to actually gate it while you breathe and all that sort of thing. So if you have to stop breathing, that's a little counterproductive. But, in any event, so they accepted me in the program. They'd had 15 people so far and the program was for 15 people, by the way, so I thought, oops, I'm out. It turned out he'd gotten permission from the NIH to continue using it before the Phase II study started in four months. These guys were thinking of people and so he <laughs> really-- it was amazing because everything they did, they were talking about it, they wanted to know about me and my family and it was just so marvelous to deal with these doctors who were really so personally involved and concerned.

But I got into the program. I completed all the radiation treatment. The first 14 patients they had completed, it appeared to have eliminated the cancer in the radiation field. So it looked like it is completely successful on eliminating the primary cancer. That doesn't mean it's over, however, because, in over 80% of the cases, this cancer metastasizes, it's kind of aggressive, and so I've got to continue chemo and stuff like that. Apparently, just in the last few weeks, at the cancer society that my Dr. Ryan up there at Mass General went to, the oncologist, he learned about a new chemo treatment that is quite effective, like, 40%. So the odds here look not too bad and it looks like there's a chance that I may actually have it cured. When I talk about life expectancy, they say, "Yeah, you've probably got, you know, two to three years maybe and you may be lucky and go longer than that."

But the rule basically is to take it a month at a time and that's essentially what I'm doing. So I'm in that treatment. I'm in the chemotherapy. I went out- I'm a jogger, that's why I'm alive, I guess, and I went out and I had a jog and a walk yesterday. I don't have a whole lot of energy, it's kind of tiring, but I'm feeling fine. I'm very optimistic. I just went out last week and bought a baby grand piano. I'm taking piano lessons and starting to work on another book and so, you know, you just go ahead and do what you're going to do. So that's what [tape glitch]

Booch: Thank you. That's great. You're truly a renaissance man. You're learning to play the piano at your age. That's great. Have you played before?

Humphrey: I sort of fiddled with it but I never had lessons really. I'd had violin lessons as a kid. My older brother got piano lessons and, after I practiced the violin for about, it must have been six or seven years, I finally concluded I had to practice an hour a day just to be a lousy violinist and so I quit it. I've always wanted to play the piano but, the problem is, as I got older, I hate to sit down and practice and, you know, you've got to repeat things ten times and all that kind of stuff, which is kind of annoying and I'd bother

everybody and you feel kind of you don't want to do that. So I did get an electronic piano with ear phones about three years ago and I started fiddling with that and I learned about fake books and stuff. Then two years ago, I went over to the piano dealer and was asking him some questions and asked him if he knew anybody who would give me lessons and so he's giving me lessons. I started piano lessons two years ago, the day after my birthday, the day after my 80th birthday. I figured I was old enough. So I've been doing that and Barbara finally agreed that, yes, we can have a grand piano so I bought one last week and they're installing it the day before my birthday, they say, next week. So that's what we're doing. I'm having a wonderful time and, as I say, you live a day at a time.

Booch: And almost happy birthday because you'll be turning 82 on July 4th, I believe you said.

Humphrey: That's correct.

Booch: Great, great.

Humphrey: I will say, in terms of all this stuff on the radiation treatment, this particular treatment I'm getting is not public. There aren't any papers on it yet. It's totally experimental. It's the only place in the world. I'm the 16th person in the world to get this treatment. I just feel enormously fortunate to have somehow stumbled into this all but-- as the accidents that happened, I mean, if the people at Sloan-Kettering had said, you know, bring him right up, I would never have run into this.

Booch: Yeah.

Humphrey: A whole range of things have happened and my daughter and the accidental e-mail and I just sort of go down the list and say, if look at it in terms of probabilities, it's about as close to a miracle as they get. So I'm feeling totally blessed.

Booch: Very good. Well, we got to this point because you mentioned you've sort of changed your work style because of the cancer and thank you for that little side track there to discuss it. Let's go back now to where you're involved with TSP and where it's headed.

Expanding TSP Use

Humphrey: Okay. Well, the TSP itself, sort of leaving me out for the moment, just talk about where it stands, we've actually, as I indicated earlier, the very first TSP was with an Embry-Riddle team in 1996. We gradually developed it, we learned a good deal more about it. We began to put together coaching and coaching classes and stuff. I have written a whole mess of books on it since. I've written now three books on the PSP. I wrote a book on TSP coaching, coaching disciplined teams, and I wrote one on leading a TSP team.

I wrote a book on winning with software, which is sort of for executives. It's a fairly short book, couple hundred pages. It doesn't come with crayons but it's sort of close. It sort of tells stories about how this stuff works. What's interesting is that book, I actually published it in, I think, 2002, one of the Microsoft executives, called a business unit I.T. manager (BUIT), when he went on to Amazon, it said, "If you're interested in that book, why don't you try this?" So he bought my book and he read it on a plane somewhere. It's that short. You can read it quickly. And he called us up and said, "I want to do this."

So we got started with Microsoft. We've got over 1,000 people using it there. It's in and out. We're not at a senior enough level with Microsoft to be sure it'll stick. Managers keep changing and so it turns on and off and, you know, things like that. But we got it going with Intuit and the way we start a lot of these, by the way, it's either a book or a paper or a talk I've given or someone calls and wants me to come give a keynote address. At Intuit, they called me and wanted me to give a keynote address out at their technology conference. I said, "I'm happy to do it on condition that I can have an executive meeting after the address." So I went out and gave the talk and the senior V.P, Bill Eerie, a marvelous guy, and he introduced me. I gave this talk to about 500 of Intuit's top engineers in the audience.

At the end of it, Bill got up, as I say, he's the senior V.P. of technology, he got up and said, "Who'd like to do this?" About half the hands went up. Well, I'll tell you, that felt great. So we started with them and their whole QuickBooks division is now using it. It's had marvelous results. They ended up, I believe, during one period, they had a reduction in calls, after they were shipping this code, they had a reduction in calls to their help-line in India, of course, of 800,000 fewer calls in an eight-month period. These calls cost $25 each. They pay out of pocket $25 each so it saved them a lot of money. Microsoft, also, one of their guys gave, briefly, a comment at one of the conference talks and he said they've spent $3 million in all the TSP training and installation and what they pay for coaching and instructors and stuff and their savings to date were $68 million.

Booch: My goodness. And, of course, you get a percentage of that, right?

Humphrey: Oh, yeah, right. The Intuit people did one thing which I was quite interested in. They did opinion surveys and the opinion surveys actually came in much higher [tape glitch] with the TSP teams than the non-TSP teams and they added a whole bunch of quotes of what engineers say and one quote was, "After I'd used the TSP, I'm not going back to do it any other way." So we're getting that increasingly. We got Adobe started, we've got, you know, people making video games in a division, there's a division of ActiVision, I think. They used TSP on the last 13 games they produced, including Spider Man II and you've seen the Guitar Hero game?

Booch: Yes.

Humphrey: Now, that was produced with the TSP.

Booch: Marvelous.

Humphrey: And basically I asked the CEO what made them pick TSP and he said, "Our turnover had hit 17%. People were just burned out. They couldn't handle it so we decided to try the TSP and now engineers refuse to work any other way."

Lack of Academic Interest

Booch: Watts, what universities would you name that are also contributing to this and teaching courses on this process?

Humphrey: That's been the biggest disappointment. Embry-Riddle did some work. They've been very helpful but I've got essentially nothing else. They got some CMU courses. It has not been adopted as a

standard at CMU at all. They have got a class here and there and that sort of thing. The master's of software engineering course is a little bit better, they've got a little bit more and they're just beginning to do more with someone who was on our team who is now over there helping with that. So I think that they may be doing better. The problem with the teaching, it's a little bit like in a hospital or medicine. It's very hard for people to get this across and to understand how to get it across or how important it is until they've actually practiced it. I mean, learning surgery from somebody who figured out about surgery by reading an anatomy textbook is not a real way to learn how to do, you know, high-quality medicine.

And that's the case here. It's very hard to get across the power of the disciplines and what it'll do for a team member and a team unless you've really practiced it. That's the issue. So the academic community, by and large, many of them have no industrial experience. They're very bright folks, their typical approach is to read a textbook and then teach it. And it's very hard to get the real disciplines that come out of it and be able to get it across in that way. So it is not sticking at universities. It's doing a little here and there. It's spotty. There is no academic involvement from a technology point of view. No one is looking at this as an exciting technology, which is unfortunate.

So that's been a real disappointment for me. I've spent a lot of time trying to convince the universities to do something with it. I started with CMU because I figure, if I can't get CMU involved, it's very hard to go to anybody else. The CMU people are great, they're wonderful, very nice folks, but they don't have the time to do it. They're willing to listen to me if I want to teach the stuff. They're happy to have me do that so they're very responsive and helpful but, in terms of them actually picking up a spear and saying, this is the technology we're going to carry forward, that is not happening and I don't see it happening. So my sense is the academic community will come on board when they have to.

Booch: Got it.

Humphrey: Either because other countries are doing it or because industry insists. I'm afraid that's where we're going to have to go.

Booch: Very well. Thank you for sharing that.

Humphrey: But that's a frustration that's a little bit early on my frustrations.

Booch: Okay.

Humphrey: But we need them. I mean, the problem is, when the White House or anybody else turns to say, "What should we be doing with software in the future?" they put together academic committees. And they, you know, they come back with better test tools and all this kind of stuff and that isn't going to get there. So that is a major frustration. Somehow, we've got to figure out how to get through that. My concern is, am I going to live long enough? I don't think so but I'm trying to get this far enough along so maybe it can survive without me.

Booch: Very well.

Humphrey: So that's the issue there. We've got a lot of stuff with industries, many, many companies, and I could probably bring some names, a mess of them, but there are whole lots of them that we've got. The

team is going strong and they're doing lots of, you know, going out helping teams and this sort of thing and so it's really very impressive what they're doing. I think they've made a lot of progress. One of the things that is happening, which is really kind of exciting, is that we've got a number of countries starting to get interested.

The first one was Mexico and they had gotten quite interested. They had me come out and give some talks and stuff and they got government support, they got the state and federal government support. The federal government, through their Ministry of the Economy, has established a group called Pro Soft to actually get Mexico [to] build a competitive software industry, outsourcing industry, in Mexico. And they decided to do it with the TSP because they wanted to be higher quality, more predictable. They could not go for lower salaries than India or China but they thought that perhaps this was something that they could do. So we've started with them.

Their strategy was to get faculty trained in this stuff and use the universities, but they decided to do it in what I consider to be a really creative approach. So we went in and trained a whole mess of faculty on the PSP and we trained them to be instructors and coaches and so they are teaching essentially our industry grade courses to students. They're also now working and have been working with companies to go out and get the companies starting to use the TSP. One of the companies doing that is called Softtek. Softtek turns out to be the largest software company in Latin America, it's headquartered in Monterey, Mexico. I work extensively with the University of Monterey.

Booch: If I may ask, Watts. Is this the same Softtek that was involved in some of the early software work?

Humphrey: I do not know.

Booch: Okay.

Humphrey: It's possible. Was that Softtek in Mexico?

Booch: There was a Softtek in a U.S. and they were involved in the, well, many things in the U.S. They built compilers, I think they were one of the proposal teams for the first four languages, just trying to make a connection. They're spelled S-O-F-T-E-C-H. So it might be a division of them.

Humphrey: No, I don't think so. This is S-O-F-[-T]-T-E-K.

Booch: Ah, a different company then.

Humphrey: Yeah. They are headquartered in Monterey, Mexico. They've got branches in Mexico City, in Canada, in Brazil and they're growing. They've got some offices in the U.S. but not much. They got several thousand people and they have launched-- as a matter of fact, they did one project with a big department store in Mexico City and what's interesting about this project is that this big department store, they needed this project to run their business but what they wanted to do was to be able to take, when a customer comes into the store who wanted to buy a refrigerator or whatever else, they wanted to be able to go ahead and take the order and give a delivery date wherever in their system that refrigerator was. So they could find it. If they had it in a warehouse somewhere or in another store somewhere, they could find

it anywhere in their system. They could get their shipping stuff and they could schedule a delivery right then.

Apparently, with all of the stuff that went on, there was a lot of, apparently, a lot of complications in putting this together, but they had actually had three separate contracts to do this. One was with a big company that you and I both know very well <laughter> and others. They had three contracts, each one of which ended up failing. Now, why they failed, I've never really gone into that but my guess is their costs started to go out of bounds, they began to run into schedule problems, et cetera, but, in each case, they had all kinds of problems. And so Softtek went to this company and said, "We can do it with the TSP." Well, they ended up doing it. The customer really desperately needed the product and, when I was down in Mexico last, I met with the team, and they were delivering the first release on schedule. The company has said, "We're not going to do software work with anybody else from now on." So, I mean, it's amazing. They've done a great job. They also told me that the turnover on their TSP teams is a quarter of that on the other teams. So their people love it, they stay there and it's very effective.

So that's the kind of thing we're running into. I've got to say, we've got Adobe, EDS is working with us, Oracle. We've got IBM Mexico, by the way, is working with us. We've got a Sun Microsystems effort, Fuji Film, Toshiba. I mentioned Intuit, Microsoft and I've got a list, I won't count the numbers, but it must be about 50 companies here, some in Europe, in the U.S, some people in India, LG Electronics in Korea. We've got a NASA group going. So we've got, you know, Siemens, an enormous list. So, in any event, that gives you a sense of what we're doing there.

So basically, in terms of where we are, we've got, when you look at the data we've got and the results. We've got results with companies, as I say, dozens and dozens. I think it's well over 100 companies now we've worked with. I had data in my last book, I was going through a data set to go and analyze stuff and the data I had was on 30,000 programs written in the PSP. So when people started questioning, you know, what do you mean? What are defect levels and things like that, I've got an enormous amount of data here. I mean, I've got data on thousands of students who have taken the PSP course and we've got before and after data, you can see how their quality work helps and what happens to their productivity, the accuracy of their estimating.

International TSP Use

So there's no question this stuff works and, as I say, what's frustrating to me is with all of that data and all the absolutely rock solid evidence we've got, why is it that people aren't just saying, "Oh, yeah, we've gotta do this and we're not." And so that gets me to another one, which is one of the things I wanted to talk about. This sort of talks about Mexico and stuff. Some of the other countries, by the way, we're working with, we've got interest from Turkey, some companies and the national government there that is starting to work with us. South Africa, actually, it's quite an effort there. There's a major effort under way in Columbia, South America. They had our folks down there and they want to launch a big effort. We've got something in Mauritius, I mean, so it's kind of popping up in a lot of places and a lot of these countries want to get in on the act. The software industry is a very attractive industry and, by and large, most of them know that you literally can't undercut the cost of labor. If people really want to buy by the hour, India and China are probably lower than most everybody else.

Maybe some of the African countries may be able to get lower rates at some point but I think everybody's beginning to realize that the Toyota strategy is a hell of a lot better than the China strategy in terms of

really getting in there and building an industry. Build a quality business, build on performing and doing what the customer wants and, as you probably know, Toyota got way ahead of U.S. industries from a quality perspective and while it took a U.S. industry about 20 years to wake up, and once they got started, they still haven't caught up. Toyota keeps improving and the other people can't catch them and that's why GM is now a smaller company than Toyota, which is kind of astounding. These people, basically, that's a history that they see and they say, "Hey, maybe we can do that here." That's what we're beginning to see.

As I say, I'm terribly frustrated that there is essentially no interest in that in the US. Even the academic side, the government side, industries are all looking at it very narrowly. And I had some experiences with a couple of big companies, one of which we know well. Another one, also the big software contracting companies that go out, and they do body shop work. They do contract software work. They run facilities. And basically, I talked to them, and the reaction that I got was-- these are from senior executives in both companies-- the reaction I got from them was, basically, they asked me, "You mean you want me to spend profit dollars to cut revenue?" And that's staggering. Well obviously, when you do quality work, you do it right the first time, people don't pay you for doing it wrong and then fixing it. So it actually does reduce revenue, there's no question

But I contend that any executive worth his salt, or any business that wants to look at the future, the choice they've got is whether they want to be the GM down the road, or the Toyota. And I think the choice is pretty clear cut, unless you've really got very short-term vision. And so that is, as I say, terribly frustrating, and what's most frustrating is, I get that from DOD contractors as well, these great big companies. I've gotten this from several of them. They basically say, "We can't afford to do it." I think the lower level managers, that's true. The lower level management really can't afford to take money out of their profit pocket and actually end up better controlling their costs, cutting their revenue stream and that sort of thing, even though it would improve their competitiveness and certainly help the national economy, and certainly the defense business, they just aren't willing to go into it, and they don't feel they can fight it through their own management. And that's a real risk and a real problem for the future of our country.

So that gets me through that one. Let me move to another one connected to it, and that's sort of the cost plus business of software contracting. There's another one I'd like to talk to, which is the large systems business. And maybe before I get to that, I ought to talk about why TSP really worked. So I was thinking of shifting gears right now. This is sort of the end of the standard TSP discussion. I've sort of brought you up to speed. And before you move off that, do you have any other questions on the status of the TSP, before I move on to talking about why TSP works and then talking to really big systems issues, and then things like that? Are you there?

Booch: Yes, I am. That sounds great.

Knowledge Work

Humphrey: Okay, so I'll just move on. The question of why the TSP works is an interesting one. You've, I'm sure, heard of Peter Drucker.

Booch: Oh yes.

Humphrey: Died a couple of years ago. I've heard him talk on a number of occasions. I've thoroughly enjoyed his books and papers. And one of the things he has said, and he's talked about it for a number of years now, is knowledge work. And he talks about the knowledge workers. And he makes a point, which I think is enormously perceptive. He fundamentally says you can't manage knowledge workers. They have to manage themselves.

Well, what's astounding to me is, first, that's a terribly perceptive point, and second, no one has picked up on it anywhere. Everywhere you turn someone says, “Sure, that's true. They've got to manage themselves. Let's have them manage themselves. Great!” But the knowledge workers don’t know how to do that. So it's a fundamental change in the whole management system to do that. And Peter Drucker clearly thought through that, but he wasn't in a position to go do it and make it happen. And so that's the question that I have been struggling with the TSP and the PSP. The question I had, and I think I mentioned when I started with the CMM and then moved on to the PSP and TSP, was to look at how would individual developers really do software if they did it right? And a big part of that was learning to be a personal manager, learning to make your own schedule, track your own progress, to manage the quality of your own work, to make commitments and to consistently meet them.

And how could individuals do that? And so that's the framework on which the TSP and PSP are built, giving people the tools to do that. And as you probably know, most software folk, you tell them to do something, unless they really know how to do it, they'll probably put it off. As I think I mentioned in my IBM story, the software people had an enormous number of things they had to do, but there were only two they really must do before they could ship a product. That was code and test. So they never got around to anything else. That's exactly true of the software community. You tell them, “What you really need to do is get your requirements nailed down and do inspections and do all this and that.” And the question most of them have is, “Well how do I do that?” They by and large don't know. They've never gone through that. They don't have a framework to do that. They don't have a process or guidance on how to do it. Hard to find people who can. So they start off coding. I mean, they start off trying to put together a design. They start doing what they know how to do. They don't have time to play around.

And that's the issue we run into in the software business. The people really are so busy, they're under such enormous pressure, that they just can't take the time to figure out how to do something. And that's very obvious. I mean, I sit down and I've got all kinds of data on how long it takes to develop a process, and the TSP process that I developed, with all the guidelines and the scripts and stuff like that was several hundred hours of work. And I'd been developing processes for years. Software developers just don't have the time to do that. They'll use what they can get their hands on. They're very pragmatic. They'll use my process if they know about it, and they're convinced it will work. And that's why the PSP, by the way, which I developed-- and every time I wrote a program with it, I would sort of modify it and update it, so when I write programs, I want to change the instruction set. When I work and use a process, I like to fix it and adjust it. I don't change it on that particular project, but I'll make notes and I'll go back and fix it.

And so I'm constantly evolving my process, but I find engineers don't do that. By and large, they use whatever process they have got. If it works, they're happy and they use it. So we learn from it when people use our processes, and we update them to make them more convenient. But the developers won't do it. They aren't willing to take the time. They don't have the time. And so fundamentally, this whole idea that “just tell the developers to manage themselves,” managing themselves takes a lot of work. People have got to be able to find out what their requirements are. They've got to understand what management's goals are. They've got to put together an overall plan and strategy for how they're going to

do it. They've got to negotiate their schedules and commitments with management, all of that stuff. Then they've got to do the work and they've got to track their progress, they've got to handle it.

There's an enormous number of things, and without guidance on how to do that stuff, they can't do it. And by the way, they also need room from management. Management has got to really take the time to meet with these people and say, “Here's what we want and why.” And that's what the launch process does. And it's extraordinarily effective. And so the whole idea of it is to actually give the developers and their teams the power to figure out what it takes to do the job. And so when management assigns a project to a team, the team goes off and takes several days to put together a plan to do it, and they really crawl through it and they do a thorough job. They've got historical data, either on their work or other teams, and stuff like that. And they'll go through and make a plan.

More Example Projects

And when they've finished, they know what it'll take to do the job and they have so much confidence in what they've done, and so much conviction, that when they present their story to management, management doesn't argue with them. I mean, basically, these people know. They've got the conviction. Management is universally impressed with the teams when they come back. And they've got credibility. And so they negotiate with management. And the teams always end up winning, and they've done that time and again. I remember one case, this was a good many years ago now, it was a team at Ford, the Ford Motor Company. It was a small team. I don't know if they're still doing it or not. I think they've got some people that may be, but I'm not sure. But I haven't been in touch with them for years.

But in any event, this team had started a project, I think there were five engineers on it. Management had gone through with them and given them the goals, the goals where you had to get this done in a year. Well, it turned out to be an enormously ambitious project, and the team basically went through it-- the management reason you had to have it in a year was, they had a market window and they had cost limitations. And the team went through this thing and came back and presented the story to management and said, “Look, with this team, this will take five years. If you want to put 20 some people on it, we can probably bring it in a lot closer, but no way we can get to a year, and here's what we've got to do.”

And they went through the story and that sort of thing, and so management said, “Let's think about that,” and then they canceled the project. Well, that's an enormous success. You think, normally the team-- and this is what teams typically do. Management tells them they've got to do it in a year. “Okay, chief.” Then the team will say: We'll try.” And they'll break their hump for about eight or nine months, and the management says, “Where are you?” and they'll come back, “Oh, it'll be another six or eight months.” And so the delay will keep going and they'll lengthen the schedule. After about two and a half years, they will have gotten where they're working under enormous pressure. They don't do the work right. There are all kinds of problems. Then they kill the project, and they waste a lot of money, they waste a lot of the engineers' time. Nobody wants to do that.

So having reality on the table right away, by and large, with very few exceptions, that works. We do run into a few cases. I remember one case in one company, which I won't name, the senior vice president had basically given a directive: “You'll get this project out by this date.” And the lab manager, I was there talking to him and with the team, and the management team. And they were scared to death, because this VP was unwilling to come to the lab and talk to the team. He was unwilling to listen to what they were

doing. It was all done remotely. And the last time somebody had gone back to him and told him they couldn't do it, they fired him.

So they finally concluded there was no way they were going to do this. This guy was dealing through a pipe. He didn't have the guts or self-confidence or something to deal with his people directly and he was totally unreasonable. And so they did. They basically did not use the TSP, and about a year and a half later, the lab director was fired, and so was the VP. I mean, this sort of thing goes on all the time. It's crazy. And so you do get a few nuts like that in the management chain, but not too many. Most of them are smart enough to know that when the team really says they can't do it on this schedule, that's probably true. The problem is that few teams know enough to say they can't do it on management's schedule. If they did, then we wouldn't have so many disasters.

So the point I'm making here is that the teams discover that when they have data and they have facts, they can talk to anybody. I mean, we've had cases where brand new engineers, with a year or two of experience, going through this stuff, will explain to a senior vice president something, and the senior VP buys it, because the engineers have the data and they understand what they're talking about, and they've got conviction. And so what these engineers discover is, they've got enormous power that they never had before, and they're able to actually negotiate schedules that make sense. Their work is a hell of a lot more fun. They work at a reasonable schedule, and it works. It works extraordinarily well.

Let me talk about one case in India, one of the Microsoft projects. They'd done a great job and I was there reviewing their results. And it turned out when they'd launched the project, they had just come off a previous project which was an absolute disaster. They worked 70-hour weeks, they were late. The project had all kinds of problems in test. It was a mess. And so they decided to learn the PSP and do the TSP on the next project. So this was the first TSP project that they'd done, the very first one.

And so in the launch, the team had gone through and said, "We've got a team goal of working 40-hour weeks. We're also obviously going to meet the schedule and all that sort of stuff," but they had a team goal of working 40-hour weeks. And so I was reviewing the results of the project. They said, "We didn't quite make our goal. We worked 45-hour weeks on average, but it was a hell of an improvement over 70." They were home for dinner. It was an experience they enjoyed. It was great. They delivered on schedule to test. The product sailed through test without any particular problems at all, and the team was just terribly excited about it.

And so we see engineers, software folk, all of a sudden becoming heroes instead of bums. And these people deserve to be heroes. They're bright, they're capable. They do marvelous work, but at the moment, they are suffering under this cloud, because they really don't have the skills to manage themselves.

And the reason this is an important issue is because knowledge work-- this is back to Peter Drucker-- knowledge work, the first real large-scale knowledge work was in software. And I'd always wondered, years and years ago, when I started, I'd done all the hardware stuff. I could manage 30, 40, 50 people doing hardware work. I could walk around. I knew where things were. I didn't have any problems. But even with a team of ten software people, unless I was really willing to take the time and really sit down and go through with each one what was going on, there's no way I could know what they were doing.

And furthermore, as Fred Brooks once said, they're 90 percent through coding most of the time. The software people have no idea where they are typically today. So you go ask them where you are, and they'll go, “Oh yeah, I'm on schedule. I'll be in to test shortly,” and that sort of thing, and they literally don't know when they'll be done. So all of that, the lack of skills and all of that, we end up with people who have no credibility with their management. They end up getting pushed into commitments they can't meet, they're failures. They'll work god-awful hours and we see that throughout the field. It is not the kind of industry that is going to be healthy and growing over time, the way it's being run today. Do you have any more questions on that? I was going to move on to--

Booch: I do have one question, and that is, you spoke about not really being able to know when they're going to be done. That triggers a thought with regard to Barry Boehm and his work on economics. I'm wondering what connection you might have with Barry's work.

Humphrey: Oh, I've been involved with Barry. I know him well. Certainly his book on software economics, I've used that. At the beginning, I used it in writing my Managing the Software Process book 20 years ago when I first joined SEI. So I've enormous regard for Barry, and actually, he has had some people teach the PSP in his program. I'm told he's familiar with it, although again, the same thing, they don't have the same kind of commitment to it that we do, because they don't have that experience. So Barry has been a supporter, but as I say, he's not really on the inside making it go. He's got other things he's committed to. Okay?

Booch: Got it. Very good, thanks.

The Boeing B2 System

Humphrey: Let me move now to large systems. I mentioned one of the problems we've had is that software has been hard to manage since the very beginning, and I ran into that very early on. I'd go out and talk to software teams. They didn't know where they were, they couldn't tell me where they were. I couldn't tell by looking at them what was going on. It was just sort of this fog of “What's happening?” and you sort of hope something will come out some day. And that's basically what we were going through. And I was able at IBM to deal with that when we put in place some structured steps and a process, so I had some idea where they were. But it wasn't at the level we had to reach with the PSP and TSP, to really understand status.

The problem you have on large programs, and it’s the point that Fred Brooks made, it was a wonderful point, it’s in his “Mythical Man-Month” book-- and if anybody hasn't read it, they need to read it. That's a marvelous book-- but he says schedules slip a day at a time, and that's absolutely true. And so the question is, when the schedule is slipping, how long does it take to know? If you're really slipping the schedule, when can you take action to recover? Today, the software community basically knows we're in real trouble when they're not into test when they should be. That's usually a year or more after they start. They're way down the road. It's unrecoverable.

The issue is, how can you detect a one day slip that same day? What's interesting is that, with the TSP, we can do it. So that's what we do. That's why they meet schedules, that's why they can come in on time. We had one team, this was a Boeing team years ago. I won't go through the story, because there were disasters with them. Half of these projects have disasters, because of management changes or something else -- all kinds of stuff that have nothing to do with what we're doing with the process. So we

run into this and it's one of my frustrations, the dynamics of the management system. Maintaining continuity in an organization in any kind of improvement effort is not totally impossible, but almost, unless you really are dealing with the top of the organization. And we were not doing so at Boeing.

This story started in about September one year. Boeing had bid to do a major update for the weapons delivery system for the B2 bomber. They were under a subcontract from Northrop, and the Air Force had told them that they wanted a lot of features but that only a few were truly critical. Those critical features had to be completed by December in the following year for a B2 flight test. So Boeing looked over the feature list and gave the Air Force a bid to do them all.

I heard that the process people had decided to use the PSP on the project so I called them and told them they should use the TSP because the PSP by itself wasn't likely to be successful. They agreed and asked me to be the team coach for the first launch. They also trained two of their people to be local coaches.

Well when I got there and we did the launch, we did not realize that having people watch the launch could be a real problem so I agreed that they could have people sit in. The team had 18 engineers and we got through the first day and a half of the launch and had made the first preliminary estimates for the development effort. During the launch, some of the observers went to see the program manager and told him that the development estimates were much bigger than their estimates so the schedule would not meet their December deadline. The program manager then killed the launch during the lunch break. Unfortunately, we did not have enough data then to prove that the test times would be much less than the original plan so they could still meet the schedule, so the team did not officially follow the TSP.

What then happened surprised me because the Boeing managers didn't know about the importance of the team having a plan to follow so they just went along following the plan that had been developed during the proposal. The team, however, had already produced enough of a plan to guide their work and since they had no other detailed plan for the work, they just followed the partial plan that came out of the partial launch we had done. They also had two coaches who helped them use the PSP.

In any event, this team got going, and they could see, they had a schedule, they had to be done in 15 months, and it was a hard stop. They had to be done. After three weeks with the TSP, they could see they were in trouble. They were able to lay it out, they were going to be about a month, month and a half late. And so the team actually sat down. Remember I talked about task hours earlier? The team sat down and figured out, "What can we do?" They concluded they had to go on overtime for a couple of weeks. They decided not to do it right away, because it's going into Thanksgiving and Christmas, but right after Christmas, they went on overtime for several weeks till they got back on schedule.

And they each established goals for personal task time and they tracked them. And they got their hours up and they met the schedule. They actually came in a little bit early. If they hadn't done that, they would have had no way of knowing, and that's the case on all these projects. If you're going to maintain schedules on these projects, you've got to manage every day. That means you've got to know where you are every day, and that takes precise schedules, data, and tracking, all kinds of stuff. And that can only be done by the developers themselves. That's why knowledge workers have to manage themselves.

The manager's job is to give them support, coach them and protect them and all that sort of stuff, but not to actually manage what they do. That's a key point, so think of it this way, on a big program. Take a program that's got thousands of people or hundreds. As I say, I had the OS 360 with thousands of people,

so I have had a lot of experience with great big programs. I'll tell you, I would have given my eye teeth for a process like the TSP. It just would have been a gift, because we really would have had the ability to manage what we were doing.

Booch: Watts, measure for me what a really, really large program is, versus a small one. Bjarne Stroustrup once pointed out to me that if I can't say what a thing is not, I haven't really identified what the thing is. So I'd like to understand your metrics for when you reach that threshold, that it becomes that size.

Humphrey: Before I do that, let me finish the Boeing story, OK?

Booch: Ok, then we'll come back to my question.

Humphrey: Well, the Boeing team ended up producing really high quality designs and code. The overall code ended up being much bigger than planned – as I recall over twice as big – and the effort was much more than expected because they were making a fundamental change in the system. The requirement was to take a hard coded weapon-attachment design and convert it to a table-driven design. That is a bit like replacing all the wiring and plumbing in a house without affecting the wallpaper. It's real tough.

So the team kept following the TSP and they were about five months late finishing the design. Everybody was worried. Because the design was such high quality, however, coding was pretty fast and they got into test only about two months late. Then the testing was a snap – they actually were ready for the December Air Force tests a month ahead of schedule.

You would think that this would have convinced management to keep using the TSP, but it didn't. The problem was that the contract called for a design review in December that first year and the development team didn't have the design review in their plan so they missed the date and Boeing forfeited an award fee as a result. In any event, because they missed this fee, management was severely criticized by senior management and the program was always viewed as a failure, even though it was delivered on time.

Several things happened that we could have avoided had we known what we know now. Actually, however, we learned a lot of these lessons at Boeing. The first was that, if management had trusted the team and explained the critical need for a design review in December, the team could have held one. They knew a lot about the design in December but, with the TSP, they were now spending a lot more time in design and a lot less in coding. So they could have given a high level design review in December even though the detailed designs weren't done.

Some of our early PSP data shows what I mean. I mean, during PSP training, the developers write a bunch of programs. In the first courses it was 10 programs and the data on program 1 is with their old process, the way they had always worked without the PSP. On program 1, the data we have on several hundred programmers shows that they spent less than 20% of their time in design at the beginning and over 40% at the end of the course. Design was taking twice as long. While programmers don't now generally use compilers, they use code generators which also try to generate code from whatever defective stuff the programmers submit, which is essentially the same problem we were running into with compilers.

In any event, what is surprising about this is that even though the total design and code time percentages went up but the total time actually went down. That is because the time spent in compile and test dropped from about 45% to only 25%. The compile time at the beginning was more than the design time and at the end it was much less. While these were all important, the key was that program quality was so much better that system test time dropped from a planned six months to about one month so even though they got into their internal testing late they were able to deliver to the Air Force for flight testing ahead of schedule. In a later presentation on this project, a Boeing executive said that this time cut testing time by 94%.

I also went out to visit the team in about early March and looked at some of their early data. Their plan showed that the program would have about 7,500 new and changed lines of code, and they had estimates for how much this would be in each of the system's existing modules. When I got there, they had only completed coding for a few dozen modules but, with the TSP, they had the data for the new and changed code for each module. I then assumed that the ratio for these few modules would be about the same for the rest of the program and estimated that total program size would be about 18,000 lines of code. My estimate later turned out to be pretty accurate. The TSP data is really valuable for this kind of thing and people don't seem to see how useful it can be in measuring job status or estimating job completion.

We also learned that we had to bar observers from all but the opening and closing meetings and we had to make sure that the managers better understood what this was all about. This experience led to a one-day course for executives. Actually, executives don't go to courses so we call it a seminar. We also have a three-day manager's course.

OK, that's the Boeing story, now, back to your question about measurement, OK?

Booch: OK. I was asking about your metrics for what is a really large program.]

The Large-System Problem

Humphrey: I'm talking about a program, typically it's three or four hundred or more people, typically involves multiple organizations, not necessarily different companies, but certainly different laboratories, different teams, different groups. Typically, they're remote and they're typically building a fairly big product, usually in multiple releases, not always. But that's what I'm talking about. As I say, it's typically a fairly large product. I won't put it in terms of software lines of code, because many of these are not software systems. They have software in them, but they're other systems. You work on these big nuclear power plant things. We've had some involvement in those. I think we have a team that's actually using the TSP to design nuclear reactor power plants. But there aren't any software people on the team at all. We have requirements people doing it too.

So you can have a big team, which is just a large collection of groups that all have to interact and they all have to synchronize their work. That's the key. And so the issue now, and I ran into it at IBM, and it was one of the most serious problems we had, and we had to keep the managers right on the ball, because we ran into what we call the "last liar problem." Fundamentally, if everybody's in trouble on a project, no one wants to admit it, because no one really knows what's going on. They're all sort of in trouble, they can see intuitively, "We're late," and they wait for somebody to have problems that are so visible, they can no

longer be concealed. And then everybody else can relax, because the first ones to admit to problems are the ones that did it. But everybody's in trouble.

And so the real issue with these great big systems is that no one really knows precisely where they are. No one really has a way of tracking a day at a time where they are. The interactions and the connections among the groups are very hard to manage, because people really can't predict exactly when they'll be done. And so all of these great big systems are enormous interconnected things. You have this big network of commitments and that sort of thing and in these great big systems, there's several things that are serious problems, and one is, no one really knows where any of the pieces stand. That means that everybody, when they're talking about their status, is defensive. They're sort of guessing.

So the individual engineer, the team leader goes to the individual engineer/developer and says, “Where are you?” He says, “I'm almost done, Chief.” Well, the team leader knows he isn't almost done and he tries to poke at it. So they have kind of a guess at when you're going to get into test, but no one knows. So the team leader’s kind of uneasy about it. He doesn't have this feeling of confidence, and he gets that from all his team. And when he talks to his management, he then gives kind of a fluffy answer and no one is convinced he's right. He knows he's sort of soft and he hasn't met schedules before, so why would they believe him this time? And as you begin to build that up, a layer at a time, you get all the way up in the organization and everybody is being defensive. No one is admitting exactly where they are. No one is sitting down and saying, “Here are the facts. Let's fix it. Let's get in and resolve the problems.”

And so these great big systems, you get this kind of defensive structure, all the way up. From the management of these big companies, it goes to the Department of Defense. And from the Department of Defense it goes to Congress. So no one knows what they're talking about at every level and the reason is because they don't start with a solid foundation at the very beginning. The engineers, the individual developers, don't know exactly where they are and if they don't, no one above them can, and the entire system is guessing.

My point is that on these great big systems, they're so sophisticated and so interconnected, that you're literally taking all kinds of risks when you do this. If you count on good luck, you're not going to get it, and that's exactly the case with these systems. That's why, by and large, these great big systems are enormously late and way over budget. The way you can deal with it is to start all the way at the bottom with real precise control, put self directed teams in place, begin to use data to track and manage it. And we know exactly how to do that. That's what's so frustrating to me, because I can't get anybody interested.

So that's the issue we're struggling with. That's the ultra large system problem. How we move to that stuff, how do we handle this? I'm hoping we'll get there someday, but it's going to be a challenge. And if next year I'll be around to do it. But that's what we're struggling with. Is that enough on that?

Booch: Absolutely, yeah. That's great. I have some questions as we get on. Thoughts for various different kinds of programming models that we'll come back to, but I think that's a good summary.

Women in Software

Humphrey: One thing before I forget. I keep saying “he” but there are lots of very good women in software. The team lead at Boeing was a woman as was the technical lead. The problem is that not enough women go into software. We have that image of a techy culture. Sort of where everybody is glued to a screen night and day so lots of women don’t see it as a fun job. I have four daughters and they like working with people, and I think a lot of women don’t see what a marvelous people-intensive field this is and the TSP really helps with that. It’s all about people and how do you lead and motivate them and how do you negotiate and mediate. Women are often darned good at that and we need more of that talent in software.

Booch: Yeah, you’re right.

Humphrey: Okay. When I go down our list, do you want to take a break for a minute?

Booch: Only if you need to. I'm doing great.

Humphrey: Let's take about a five minute break.

Booch: Will do. I'll stop the recording now.

Booch: We're back again, and Watts has a handful more topics to cover here. Let's talk about issues of inertia. Why is it so hard to change organizations and change people?

The Change Problem

Humphrey: I've struggled with that. One of the things I keep running into, and I ran into it at IBM, when IBM was at the top of the heap, you literally couldn't get them to look at things differently. This was IBM. We were winning. Why do you want to change things? We're ahead. The conclusion I reached is, one, when organizations are at the top of the heap and they're enormously successful, it's extremely hard to get through. No one has to change, unless you run into someone who is a strategic thinker and is looking out for opportunities and ways to improve, and very few do.

There are very few who, like Art Anderson of IBM, say, “there's always room for improvement.” Most of them are basically working on what they're doing, trying to improve costs and profits and the competition, but they're rather narrowly focused on the paths they're on. So when they're winning, they don't have any pressure to change it at all.

And then the other side of that is when you get into harder times, now they get into trouble. The difficulty there is, they can't afford to change. They're under financial pressure. Then've got high-priority stuff they've got to do. They're in survival mode. And so that's what we run into. To get people to change when they don't have to is what we've got to deal with. When they really have to change, it's extremely hard to get their attention. Now we occasionally run into people like that, for instance, these foreign countries. They're moving, not because they have to changed, but because they see an opportunity, and they're after a strategic change. Well, that takes vision, and there aren't a whole lot of people with the kind of

vision to do that. And that's what Toyota did. They had a vision, and they talked to Deming and others, and they started moving in that direction. I don't see that. Very little of that comes out. Occasionally, an Adobe or an Intuit or someone like that will have a vision, a manager down in the bowels of Microsoft. He's had a vision and he's been a great supporter, but it doesn't get up to the top. We've never gotten to Ballmer and the top executives. They've got other stuff to do. When they really need it, it'll be a little too late.

And so these great big organizations, it's very hard to move them. So the lack of pressure, the lack of something requiring change, means it's very hard to do it. And when you really have to do it, software improvement is not on the priority list, because they've got to survive. Take the case of General Motors. What are you going to do there? You get in there, you go into Delphi or one of these suppliers. Every one of them could use our methods, but they're not going to pay attention to it. They really can't afford to right now. They've got a survival problem and that's it. I believe that's the problem.

Booch: So as you think of companies that are on the top of the heap now, tell me where you think Google might fit into this. What advice would you offer to them?

Humphrey: My reaction is that any company like that, certainly Google would be one of them, they really ought to sit down, look at their technology, what's going on, what works and what doesn't. Why can't we deliver products on schedule that work exactly the way we want them to work? I haven't looked at their stuff in great detail, but that's certainly true. You look at that at RIM and the Blackberry. They've got all kinds of software they come out with and quite frankly, if you look at it, they've got a hell of a lot of defects in their software. It doesn’t hit you and I too much, but it costs them a bundle. That's true in a lot of these places, and I see enormous staffs of people fixing and responding and that sort of thing.

So looking around and saying, “What's going on, and what are the technologies that really show promise and how could we improve and grow our business?” and that sort of thing, there ought to be at least some effort like that in these companies, and it ought to be from the very top of the business. The senior executives ought to be actually looking around. What are the choices? What are the alternatives? Because winning companies, by and large, ultimately lose because they're blindsided. Somebody comes at them that they didn't expect. They're just not ready. All of a sudden, there they are and they're out in the wind. That's a real problem. Now General Motors wasn't blindsided by Toyota. They saw them coming, but IBM was. IBM, I was there and we talked about that. Somebody new came in and a few years, bam, they were really in trouble. That could very well happen to Microsoft and to Google and every one of these companies. A lot of people would like to take a piece of that action.

What people keep forgetting is that the rate of change is accelerating. According to the Economist (Sept. 19, 2009) 24 firms dropped off of the Fortune 500 list, on average, every year from 1956 to 1981. From 1982 to 2006, that number jumped to 40 firms a year that got pushed out of their top spot in US industry. That is an extremely high rate, and it happens to all of the big fat-dumb-and-happy companies. Nobody gets a free ride any more. Look at the companies that are gone: the old AT&T, Pan Am, Gulf Oil, and all the others.

Booch: When you're on the top, people like to push you off of the top because they want to be on the top.

Humphrey: Oh yeah. Well, they're going to. They're going to work at it. There's enormous motivation to do it. And the thing that's interesting is, in our business, in the software-connected business, getting into

this business is really remarkably cheap. All the open source stuff, the very low cost hardware, the low cost of communication. People in Zimbabwe, when they stop all the fighting over there, I'll tell you, they've got an awful lot of very smart people. You take a look at how bright Obama is. He's a direct descendant of those folks in Africa. The people there, there are an awful lot of potentially very smart, very capable people throughout Africa who would work for peanuts, and they're looking for opportunities and growth. And this is true throughout the world. We just sit in our little comfortable corner here and assume that we're going to stay ahead. The odds are rather small. There's an enormous amount of energy behind trying to crawl past us and take us for a ride.

Booch: Would it be fair to characterize it, your view relative to the US, Africa's the next India?

Humphrey: I'm not sure. I think Latin America's going to surprise us. I think competition is coming out of there faster. In terms of the politics, I'm afraid that in Africa it’s going to be too long before they can get control of their political situation. The enormous problems they have with graft and all of that sort of thing. The corruption is really destructive.

But I think India and China still are continuing to move up. China, I wouldn't be surprised. I gave a talk to a university group, must have been a couple of hundred people, mostly students. I asked, “How many of you know about the PSP and TSP?” and just about every hand went up. So I said, “How many have actually taken a PSP course?” Almost half the hands went up. So they're looking at this stuff. The companies aren't doing it yet. They're building the skills and stuff might happen. Some of them may wake up in time. I will say, by the way, when I talked to people at the Microsoft lab in Beijing, they said they wanted to go out and hire about 100 engineers. So they put out some kind of a post and they got 10,000 applicants. The talent and the skill available around the world, are just way beyond anything we can comprehend in this country. We have typically 50,000 people will apply for our Westinghouse science scholarships in the US every year. China has a similar effort and they have 5 million applicants.

So we've got to stay awake here. If we don't really stay out front and lead the world technologically, and with our methods and our skills-- and we've got the opportunity to do it-- but if we do what the US industry did with Deming and wait 20 years till we start to pick up on these methods, we're not going to catch up. The world will be moving too fast for us, but that's what scares the hell out of me. It's frustrating, so very little strategic vision. So that's a problem.

The National Medal of Technology

Booch: Got it. Well, let me turn now to the story of you receiving the Medal of Technology. I think you had the ceremony in 2003, although you were given the award earlier than that. Can you tell me the story around that, your meeting at the White House and all that? You're only the second Medal of Technology technologist I've run across, and I'm always fascinated by those stories of how it came to be. What an honor to receive that.

Humphrey: Well, it certainly was. The award was actually the 2003 award, which I was given in March of 2005. It was delayed because the President had a whole lot going on at the time.

Booch: For the people who are listening to this, you might want to expand upon it, because you and I know what we're talking about, but there may be generations after us who may not necessarily know that context.

Humphrey: Right. Well, it came to me as a real surprise-- and let me back up a little bit. The National Medal of Technology-- and this is true of all of these medals, by the way, they don't just appear. Somebody has to nominate you, and that's an enormous job. And Bill Peterson at the SEI, who's director of the process program, took it on himself to submit the nomination. And he actually went through the whole cycle twice. The first time, apparently, it didn't fly, and he went back and did it again. This time it did. The first time I heard about it, I don't remember who I got a call from. I can look back in my notebooks, but I got a call from an assistant secretary of commerce.

He called me up and told me that I wasn't able to tell anybody yet, but I was going to receive the National Medal of Technology and he wanted to make sure that my calendar would be clear and it would be in the White House and that sort of thing. Basically, I was absolutely staggered. The National Medal of Technology was basically originally established-- it's managed out of the Department of Commerce and they sort of see it as the U.S. top technological award. There's also a National Medal of Science, so they basically go together and they have the awards together.

They view it as sort of the U.S. equivalent of the Nobel Prize. As I say, when I heard that. wow, I couldn't believe that they wanted me in that league, to have me up there. But in any event, so they put the whole thing together, and then a lot of people started to contact me with this thing, because they had to put together the program. And part of the program was a video, that there'd be a two to three minute video on each of the winners. And the program actually started-- I've forgotten the date, but it was in March-- we started-- I'm trying to remember.

The ceremony, the first ceremony, I think was in the White House. We went to-- we met at one of the hotels in downtown Washington. So we all arrived there. We arrived there and they had a dinner the night before. And so we all met and that sort of thing. It was a small thing. There weren't a whole lot of other people there. But they told me I could have three guests. I said, “Well, I've got seven kids. How do I do that?”

Booch: Who do you love the most?

Humphrey: Yeah. So they said, “Well, let us check with the White House.” So they came back and said, “You can bring them all.”

Booch: Oh, sweet.

Humphrey: This is marvelous. So they all came. Actually, one, unfortunately, was in Australia and couldn't be there. So Barbara came with seven of our kids-- so we had six of the seven and Barbara and myself. We went to the dinner at the Ritz. I think it was the Ritz.

<inaudible background comment>

Humphrey: No, the night before dinner was at the Ritz, I believe. The small dinner, it was very pleasant. Then we all went back and we had to appear the next morning, of course, all properly suited up in business attire and that sort of thing.

Booch: Watts, tell me the date for this again.

Humphrey: It was in March in 2005. I don't know if the date is on the stuff here. I've got to--

Booch: I can look it up. I can Google it while we're doing this.

Humphrey: It's there and I could go back and find it, but I don't remember the exact date. But in any event, so we arrived back at the Ritz hotel the next morning, the morning of the award, to get bussed to the White House. And the awardees all had to be there early, because we were going to get bussed separately, and then the family was going to get bussed in on their own a little bit later.

Booch: Here we go, by the way. It was on March-- gosh, I had the date here a second ago-- March 14, 2005.

Humphrey: Okay. Yes. Well, that was it. I think it was a Thursday if I recall, I'm not sure. But in any event so we got bussed over to the White House about 7:30, 8 o'clock in the morning, fairly early. And as we were riding over, the gentleman sitting next to me I started chatting with him and it turned out he was in, I believe, biology. I'm not sure if it was biology or something of that sort. And he was talking about what he had done and it turned out he had also won the Nobel Prize.

Booch: Oh my.

Humphrey: And so there were a number of people like that. And I thought wow, I'm in rather exalted company.

Booch: My recollection is that the other award winners of your time were Jan Achenbach, Bob Metcalfe who did a little thing with the Internet as I recall, was he there with you? At least according to the list he was a 2003 recipient.

Humphrey: Yes, well, I'm sure he was. I really don't remember the names of everybody who was there. But there were a few medal of technology winners, I think only about four, <inaudible> and several other gentlemen. I remember we talked about it, what they did and the people who I think had come up with the catalytic converter, two gentlemen there and several others. But I don't remember them. I didn't really have a whole lot of discussion with them but there was quite a crowd, most of them were Medal of Science winners and those were the Nobel Prize winners that I had run into.

Booch: Got it.

Humphrey: It turns out Fred Brooks and Bob Evans also won the National Medal of Technology for the 360 system which I learned later. So anyway we arrived over in the White House…

Booch: And by the way, I have that story of Fred at the White House. Steve Jobs was there at the same time. So we have a background on that too, which is why I'm fascinated with your story. So you arrived at the White House.

Humphrey: So we arrived and they bused us into the West Wing through sort of a special entrance there and then they walked us all in and we went into a special room in the West Wing on the first floor and we were to stand there and then in came the president. And he walked around and greeted each of us and chatted with us and then he left. I mean he just came in briefly and greeted each of us. This is ahead of the ceremony. And then we waited there and basically all of the guests and visitors and stuff went into, I think it was the East Room in the White House which is the East Wing, not the West Wing. And it was the wing closest to the Treasury. Remember, I talked about visiting my uncle in the Treasury? It was right there. So yes, we waited there and then finally when they had the whole thing all set up and the room, everybody in the room et cetera they came and called us and we went in order and a marine band was playing and everybody stood and applauded. I mean it was hard to keep your cool.

Booch: And your children were in the audience then, too.

Humphrey: They were all there. My wife was there. And it was emotional. I mean it was really a very powerful period. So we all filed in and sat in the front two rows and then we all—- everybody sat down and waited for a few minutes and I think there were some comments made by the Secretary of Commerce, I believe, to greet us and that sort of thing. And then "Hail to the Chief" and we all stood up. So the president came in and went up on to the platform and he had, I think, it was a Marine officer who was handing him the stuff. And so our name would be called. Somebody would read the award and we would come up and stand beside the president and there would be a bunch of pictures. And the he'd turn to this Marine officer who would give him the medal with the ribbon and everything and he'd take and turn and he put the ribbon over my neck and then he handed me the plaque. And we stood and got a picture with the president and then we shook hands and I went back and sat down.

So we went through the whole thing. Each of us got a brief discussion. The president, I didn't say this, the president started with a brief talk about the national medal and what it was and how important it was to the nation and that sort of thing and he gave a very nice talk. It was very brief. But we went through the whole ceremony he gave an award to each of us and we all got our medals and went and sat down and then the president left. And then we all filed out, again, to music. And went back out and there was a reception and then everybody else came out and we all got together and there was a luncheon and all kinds of-- it was really a gorgeous luncheon and they had all of this stuff there. And it turns out my congresswoman was there and a whole bunch of people showed up. My congresswoman from Sarasota, Florida showed up as did a bunch of other people. Some senators were there. It was quite a crowd and a very impressive group. And lots of pictures and that sort of thing. Of course, we got pictures of me with the family and with the kids and all milling around there.

And then after some time we left and went back. We had basically the afternoon and we didn't have much to do but that night there was a formal dinner and reception. And we had to be all in tux and evening dress and that sort of thing and all of our kids did too so we men had rented tuxes for the occasion. I don't own one any more, not used that often. So we all got dressed up and went to this reception. It was really a fabulous event. And it was chaired by the Secretary of Commerce who opened. And before the dinner, again, we waited outside and when the thing was all ready and everybody was seated, the Marine Band played and we filed in and stood in front of the dais by the Secretary of Commerce. And I think he made

some comments to each of us when we walked up. I don't know what we got, we walked up and shook his hand anyway, but I don't remember exactly what else happened.

But he had a discussion with each of us and then we sat down and then, after we had eaten, they went to a video where they played about a three minute video of each of our careers. And at this point let me break and describe how they put that together. I couldn't believe it. They had a group that they had hired who did this. And they called me because what they wanted to do was to interview me on the phone but they wanted a high quality recording. They also wanted a whole bunch of pictures and they had some questions. And it turned out at the time they wanted to interview me I was out at Microsoft. And I told them that so they got hold of a Microsoft VP who's apparently on their industrial award committee and he arranged for the--Microsoft has a research with the sound lab out there--so he arranged for me to go sit in the Microsoft sound lab with all of the microphones and stuff so it's a very high quality recording they set up and these people called me on the phone and they asked me a whole bunch of questions.

And so you know I just answered them and it was just a very kind of a casual conversation for about 30 to 40 minutes. I had sent them a bunch of pictures that I'd found of albums and stuff when I was kid or in college and family stuff, work stuff and stuff at SEI. And it turned out that SEI had sent them some videotape and some pictures and stuff. IBM had also, they pulled out stuff out of their archives. So these people did an extraordinary job and they had about two weeks to put this together. And they put together this absolutely beautifully polished two to three minute video on me. What's amazing is they did it on everybody. And so you know I don't know how they did it but it was an extraordinary piece of work at really high speed. I mean they did it just very quickly because the whole, as I say, the whole thing this ceremony had been put off, put off, and put off and then finally the president had a window in his calendar and they said, "Okay, we can do it in like two-and-a-half weeks." So they only had a couple of weeks to put this whole thing together. They hadn't even told us about it. And so as I say these people did an amazing job and it was really quite a video. I've got a CD of it and my plan is to send it with my stuff to the museum.

Booch: That would be grand.

Humphrey: And I think I'll even send them my medal and the plaque if they want it. I guess that they might.

Booch: I think there are very few people at the museum in their archives that have such a medal.

Humphrey: As I say, I must admit I'm thoroughly blessed. And while my kids would love to, they've all seen it, but I think it would be much better to have it in a museum than have it in somebody's home.

Booch: Sure. Well, thank you. That's a wonderful story. Do you have some follow up on that? I'm sorry. I didn't mean to interrupt.

Humphrey: I'm not quite done. So then they had the dinner and the next day there was a session over with the Secretary of Commerce where we each gave about a 10-minute talk on what we were doing and that sort of thing. And again, they presented us with a big sort of a placard board that described the medal and all of that sort of thing and described our work. And then that was the end of it. But it was just a marvelous event. And I was so delighted and it was marvelous. I could bring all of our kids and Barbara. I

was sorry to miss one of them but we had a great time. But it really was a marvelous experience personally.

Booch: That's marvelous. That's wonderful. That's wonderful. What a great story. I'd like to turn our discussion to a couple of remaining topics. You've had such a vast and colorful history here I'm curious as to your take on a few things that have been popping up in the software space here in the most recent years, in particular, your take on extreme and agile programming, what you're thinking about open source development. I believe you have some opinions with regards to languages as well. Can you comment on this?

Agile Methods

Humphrey: Sure. Let me talk first of all about agile programming. I actually went to Agile Universe a number of years ago. They asked me to be a keynote speaker. And so I was there and listened to a whole lot of talks, talked to a lot of people. And I have just a very general reaction to a lot of the agile programming and the agile movement. There's some very positive elements to it. I mean it really is looking at not just coding but it's looking at teams and how do you do programming. And so none of the other methods have really done that. And so this is really thinking about the dynamics of how teams do projects. And I think that's very positive. And I think they've done a lot in terms of applying discipline. People think about agile as sort of something simple you can throw in but if you really go through the methods whether you go through SCRUM or XP or whatever, they're disciplined methods. They're not just things people just go in and throw in something.

And the problem that I was concerned about and I run into time after time, is that people don't use them in a disciplined way. They tend to sort of pick up and use the parts they want. I mean people say they're using extreme programming and you really poke at it and you discover that they're not really doing re-factoring and they're not really doing this or that but they're not doing design work. And so that's the part of extreme programming they picked up on - not doing design. And so it really isn't a disciplined framework that people use in that sense. It's-- potentially if they did it right it would be. We, for instance, had an issue with Intuit. They had a lot of SCRUM teams at Intuit. And a study they did -- a fellow named Jim Sartain, by the way, and he's been a great supporter. He's a great believer in data. And he was the leader of the TSP effort at Intuit. And he subsequently left. He's at Adobe now. And he brought the TSP there as well. So they got a big effort.

But in any event they were looking at the data on their SCRUM teams and how they were doing. And based on everything that he could determine, the SCRUM team's performance was not as good as the non-SCRUM team performance. It's nothing to do with TSP or anything else. The SCRUM did not appear to be greatly improving the performance of their teams which kind of surprises me because I believe a lot of the agile programming methods really do help. My only conclusion is that the Intuit people probably had some pretty disciplined efforts already underway. And as a consequence this was not a big plus. And I don't know if you know but Intuit and the tax business, they have to have an annual cycle and a very tight plan so they really do manage pretty tightly to get the stuff out on schedule, predictably because the world can't wait for that. The tax stuff has got to be there when it's needed, period. <audio glitch> And so they were doing pretty well and apparently the SCRUM wasn't helping. And so they brought in, is it Ken Schwaber.

Booch: Yes.

Humphrey: They brought him in and he basically went through it and said that it was only about one or two of the teams were really doing it right. So they started to focus on that. And in parallel they started introducing the TSP. And it turned out that a lot of the TSP teams had used SCRUM before and wanted to use it with the TSP. And so they were using SCRUM and the TSP together. And there's no reason you can't. SCRUM is basically an approach on how you do stuff. And the TSP gives you a whole series of specific things you do behind it, the TSP launch works fine, all of the measurements work fine, the quality management works fine. So there are a few places where we've got some rough edges to make it fit but not a whole lot and that's true of any of these Agile methods. And so they're not-- they don't conflict with the TSP. One of the concerns I've got is that people seem to see them as conflicting.

And that was one of the big problems I had with the book that Barry Boehm and I think it was Taylor that put a book out on discipline versus agility or something. And he was basically putting us at extremes that you're either using the PSP which is a discipline extreme or using Agile which is the flexibility extreme and that's totally misleading. I mean they're not opposite extremes, they're complementary. And so I think there's an awful lot of misleading stuff in the field about how these things work. And my reaction is: the practices in the software business are so bad today that any orderly method will improve things. And I think the agile methods will do that and that's fine. Unfortunately, they don't go as far as I believe they should to provide the real improvements we need. And I think I told you before, the level we need to get to and it isn't instantly but it's going to be not too far down the road, we need to be talking about a few defects per million lines of code. And that's a factor of a thousand better than the quality we're getting today. And you don't get that by trying harder. And that's basically all of the stuff we're talking about, all of these other methods are to use tools and bright people and have them try harder.

And my contention is, that's certainly great. Use the best people you can get. Use the best tools you can get. I've got nothing against that. But you've got to use a very disciplined framework. You've got to have data. You've got to analyze it. And you've really got to work with and focus on, how do you measure the performance of your process and how it works and the tools and your people and what they do. And that way we can move forward. And we, as I say, with the TSP we've gotten vastly better performance than I had ever expected. A third of our users, one of the studies found that a third of those people, a third of the projects, delivered products that had no defects found by the users, period. Now, that's a third of them. I mean that's an enormous number. I think that's probably not holding up now because the early ones were all ones that we coached. We're getting more and more that other people are coaching and they don't have quite the background we do and all of that sort of thing. So the numbers aren't quite as good. They're a lot better than average but not as good as our teams.

But nonetheless, the programming community is a very sharp bunch of people and they can do extraordinary work. The thing that I like to say is that human performance is unlimited. I look at one thing, I don't know if you've ever looked at it, but the time it takes to run a mile if you look at the data, race data, I've it got here. I've got an Excel spreadsheet I can bring up. The time it takes to run a mile, world record. Yes, in 1865 it was four and half minutes. And they thought no one would ever run faster. Then in 1937 it was 4.07. Then in '54 they broke the 4-minute mile. And in 2000--2009 it has gone down to about 3.7 or something. I don't have the latest numbers but it's 3.7, 3-point-something minutes now. But if you take that and make a linear regression projection of how fast a man is going to run in the future, we will get to a 3-minute mile in 2116. And a two and a half minute mile in 2200. Well I'm not sure we'll ever do that. Linear regressions are probably not likely to happen here but it's extraordinary. But it's been a linear rate of improvement now for well over 150 years almost. It's been improving linearly all along. And it's quite amazing. So human performance is extraordinary.

Booch: So is your running a mile going to fit on that list some place?

Humphrey: No. I'm off the scale. But I will say if you compare that, and I did, with the Kentucky Derby, the rate at which horses run. And Big Brown ran the Kentucky Derby in what was it, 2.03, I think. Let me see if I got the time here. Yes, the time in minutes. And yes, that was in 2008, 2.03 and what's interesting is if you look at that and you look at the curve for the Kentucky Derby for over 100 years, it's flat. I mean very slightly better but not much. I mean there is no measured improvement. And they've got the advantage of breeding. They've got all of the technology advances, everything else. And horses aren't getting faster. And so my point is the people are running faster, and it's not that they're running faster because all of these other things are helping them, it's because people are learning how to improve themselves, the whole motivation, and the whole improvement framework. People can just do extraordinary stuff. World records are broken all of the time. And how can that be? This is a world record. It stood for years. Johnny Weissmuller once said-- he died-- I don't know if you remember Johnny Weissmuller…

Booch: He was the old Tarzan guy.

Humphrey: The old Tarzan. In the early '30s I think he held every world record in swimming, just about. He had a whole mess of medals. And he was basically beating the world. And at the time shortly before he died in his '80s, he said that, "Today my world records are routinely being beaten by high school girls," I mean that's extraordinary. And I was over giving a talk at Embry-Riddle, talking to the students and the faculty were there. And I was talking about human performance and improvement and that sort of thing. And I told this story, I told about Johnny Weissmuller and I said, "Who here has heard of Johnny Weissmuller?" and all of the faculty sort of standing in the back all raised their hands. And one student in the middle raised her hand, a young girl. And so after I finished my talk one of the professors came up to me with this girl in tow. And she was a little thing. I mean she wasn't very big, a little over 5 feet tall. And he said, "I'd like you to meet this girl" and so I did and I shook her hand and we chatted briefly. And it turned out she was the first high school girl to break Johnny Weissmuller's world record in the butterfly. I looked at this woman. Have you ever tried to swim the butterfly?

Booch: I have. It's a terribly difficult stroke.

Humphrey: That was extraordinary. And so my reaction is that that's just extraordinary. It's just beyond what you could ever expect. People can do amazing stuff. And frequently they're not aware of what they can do. So I think that's one of the things that I think is just-- I'm not quite sure how we got here. Were we on agile still?

Booch: We were on agile and I wanted to turn then to the discussion of open source and what your thoughts are on it are.

Open Source Programming

Humphrey: Well, open source, I think is a fascinating trend. I was very interested in *The Cathedral and the Bazaar* and some of that as to what's going on. And it kind of amazed me that we've got so many people that are so talented that are willing to actually contribute to essentially [the] public good. And

they're very proud of their work. These are people who view themselves as craftsmen or -women. And they're proud of the quality of what they do. They want to get it out there. They want people to look at it.

That's what we need in the software community. It's the kind of performance that by being able to motivate that I think it's just an enormous advance. Now, the other side of it is, it's not clear to me how broad the open source movement can get. I think there are areas where it makes a lot of sense but I just can't see somebody getting an open source program to fix the problems I mentioned with this company in Mexico that wanted to sell refrigerators and stuff. No one is going to build that kind of stuff open source. And so it's a niche. But it's become a very interesting niche which I think is likely to really cause Microsoft enormous trouble. I think the Microsofts, maybe the Oracles, I don't know but people that are building the widely used, general-purpose programs I think are quite exposed. And the reason I think they're exposed is that the companies that are building them are motivated to get continuing revenue.

And to get continuing revenue they've got to either have a service system where you get in touch with us and we'll fix your problems and that means you've got to have problems. And we'll give you upgrades. That means you've got to be making upgrades and all of that sort of stuff. And so people want to buy the operating systems or get them to do a standard job on their computer. And by and large, you and I when we get an operating system, we don't want the darn thing to change. But the manufacturers and suppliers want to change it because that's how they get revenue. So they keep trying to come up with these added little gimmicks and features that will make people want to buy it. And they're not terribly successful with that because, I mean if you noticed, what was the new one, Vista?

Booch: Right.

Humphrey: Nobody moved to Vista. I certainly didn't. And the whole SEI didn't move to Vista. Nobody wanted it.

Booch: I would have pegged you a Macintosh guy.

Humphrey: No. I do have some kids that are, but most of them are-- I've got a daughter who's an artist. She runs her own art and design business and she uses both Macintosh and PC. But a couple of the kids use Macintosh but I used the PC originally. I had one of the very first PCs at IBM. I wanted to get it. And I held off writing my Managing for Innovation book until I got it. That was my book number two that I published, I wrote, I did on the PC. The first one I did on a typewriter, a portable typewriter. But I did that one on the PC. And of course the media and everything else on that is so badly out of date, I don't think I've even got the manuscript any more. But that's happened. The media has changed so many times that it's no longer compatible and you can't use it. It's frustrating.

But yes, so I stayed with that. SEI was all Macs when I got there, Macintosh. But I stuck with the PC and they ended up moving to the PCs as well. So no I'm on that. But my sense is the whole operating system business is very likely to crater out. The open source community, at least when you've got operating systems that are available and that work and particularly when Neelie Kroes in Europe is pushing them to begin to define interfaces and begin to nail stuff down they're exposed just like IBM was to the plug-compatible competitor business. And so I think the writing is on the wall, whether it's five, ten or twenty years, I don't know. But the operating system business I don't think is going to be a viable business long-term.

Booch: Got it. So tell me what you think about languages, in general, programming languages? I'm sorry, go ahead.

Computer Architecture

Humphrey: I was just going to say before I move to that I'd like to talk also about computer architecture. Because I think, quite frankly, the whole architecture of the computer and operating system today really ought to be in microcode. Bring it all the way down so you've got an interface that gets all the way out to the application environment, the API. So I believe the operating system ought to be part of the computer itself, and I think, quite frankly, the move to do that, they have to actually change the nature of the architecture. Remember I mentioned that I designed two computers when I was way back at Sylvania?

Booch: Yes.

Humphrey: One was UDOFT, and the other was MOBIDIC. And UDOFT was a very creative machine. It had two memories, one for instructions and one for the data. And I believe that's where we're going. I believe we're going to end up with architectures where the operating system and all the stuff that provides virtual systems and protection and security and all of that, is in separate memories and is totally inaccessible to anybody through software, period. We've got to get an iron-clad door, and people can't get to it, period. Most users will use the API and stay there comfortably, and you will have virtual machines and communicate between them, and ways to move data around and all that kind of stuff. And the security and the foundation of the system and its reliability and its recoverability and all of that is completely protected from all the hackers and everybody else who wanted to get in and take over our systems. And I think until we get there, we're not going to be safe. When we get there, and there's no earthly reason we can't, but I don't think we can afford to stay where we are now. The world is just using software for way too much sensitive stuff, and the level of crime-- people stealing stuff, stealing identities-- people can come in and they can dump sewage into rivers and all this kind of stuff remotely. It's frightening.

Booch: This reminds me of something you said in one of your "What's New?" columns that I picked up on. You were basically talking about what you think the future of the web is. I pulled it out, but it was something to the effect that you really didn't see the web becoming a place for pervasive computation. So I'm just trying to get clarification on what you mean, because the folks that are heading towards cloud computing and the like and software as a service-- what's your take on that? Because your comments in "What's New" seemed to say you don't think that's the way we should be going.

Humphrey: What I saw-- and I actually worked with people who were moving, going to local systems where people had terminals on their own systems and they would be able to do it remotely and this sort of thing when I was at IBM. So we were trying to build such systems where people would, instead of having their own capability, they would have to depend on a central installation. What I've found is people are willing to do that for some of their stuff, but not everything. So I think there's a range here, and that cloud computing, I think it's almost certainly got some opportunities. There are people who don't want to have a system installation and that sort of thing, and I think that's appropriate. There's certainly nothing wrong with it.

But I can't visualize putting all my data on somebody's cloud, and then all of the sudden having them introduce a new incompatible something or other. Because our field has been doing that. Every five years

or so, something new happens that's incompatible and you have got to re-do it. When you start dealing with databases, and we're talking about people with terabytes of data, we have got to have persistence of this stuff. We can no longer begin-- we can't tolerate moving to new, incompatible things.

So I have two problems with this. One, I think just trusting somebody to protect what you need when it's not in their economic interest is something we're not going to see a lot of. People are going to be very cautious, and my guess is it isn't going to happen real quick, if at all. And I don't think some of it'll happen at all. I think we'll see people using it for various things, but I think it'll have some limits. It may go further than I expect, and maybe it will. I'm not that good at forecasting the future, or haven't been. But I'm reluctant personally, and I don't think everybody's going to want to move there. There's way too much that people are unwilling to give up control over.

And the other side of it too. Unless we're going to get to a point where communication connections can't be interrupted, power lines can't be interrupted, there's no way you can-- all that sort of stuff-- all of this stuff doing everything through the web, you're basically dependent on the whole infrastructure being completely reliable. And in hurricanes and some kind of natural disasters, that isn't going to happen. So I think cloud computing may have some big possibilities and it may be a big deal, but I'm dubious about how far it's going to go. It isn't going to replace everything. Today people are still using punched cards. I mean, you go way back.

So I don't think see the old stuff going away. I see it as another option, but it will almost certainly have some benefits, and people will use it. I'm not sure how widely it's going to get used, but it may be more than I think right now. But they've got a lot of challenges to figure out how to make it really widely used, and I'm not sure they've got their arms around it yet. They have got to get quality under much better control than they have, and maybe the TSP will help with that. Certainly it would be nice if it did.

Language Trends

Booch: Very good. And then let's turn to languages. What's your take on the state of computer languages? You spoke about computer architectures, but the other side of that's a software piece. What are your thoughts on it?

Humphrey: I believe languages have been going the wrong way. I think it's kind of hard to describe, but if you just take a look at, for instance.NET, and you take a look and all of these big systems that are essentially automated, they put an awful lot of capability in them. Have you ever written a program with.NET?

Booch: I have not had the pleasure of doing so.

Humphrey: Well, it's not easy to learn, I'll tell you. I struggled with it. Basically every program you write, you're learning from scratch. You've got to go in and find out what does this mean, and what can you find that does this? Remember I mentioned earlier where you had the FAA program when Amdahl and some people took 40 percent off the size and running time of some programs? And they did that because they really had a deep understanding of the language that people were writing in. My problem is, that kind of deep understanding, the ability to think creatively in the language in which you're working, allows people to do really extraordinary work. When you get these languages that are so really complex and using it-- I

mean, for instance.NET, to use.NET you have got to deal with a database system, and I just wasn't willing to bother. I mean, I'd go into [Microsoft] Netmeeting and get one of the ladies up in Pittsburg, Noopur Davis, to help me get that part of the program done. It was way too complex.

My concern is that the language problem, the language really should be the interface between the programmer and the system, and it should provide the capability for people to use efficiently and accurately, without error, to describe what they want to do with the system. And my concern is the languages today are not designed based on understanding what is efficient for people, what things cause errors. I mean, I don't know anybody who's got good data on what are the things that make a program erroneous, and can we do anything with the language to fix that? I mean, there are dangling pointers. Why the heck do we have to screw around with dangling pointers and buffer overflows?

There are lots of issues. Over 50 percent of all of web vulnerabilities are due to buffer overflow. Why do we have those? There's no earthly reason that any language system would permit you to write programs that have a buffer overflow. So my sense is there are a whole lot of things in really thoughtfully designing a language as a translator between human beings and machines. But we haven't gotten there. The language ought to be designed in order to understand the error propensities of people, and what are the things that make them have trouble and waste their time, and that cause errors, and that are hard to test? And there are a whole range of things.

What surprises me is the whole thrust on designing languages is how do we make something that's richer and richer? It's a technology focus. We're not thinking about people. So that's where I think the whole language business is way off-kilter. It's not thinking about the people that have to write programs and what it will take for them to become truly skillful and to produce error-free work. And I think we can do it, if we really focus on making the languages really convenient for people.

Booch: Why don't you talk about the rise of all the scripting languages? Perl, PHP, Ruby-- those kinds of things.

Humphrey: Unfortunately, I haven't used them. I've heard that people love them, but I just never had the time to sit down and learn them. I basically taught myself all the languages that I've used. I never took a course in a programming language, except the very first one I did at MIT. That's where I met Barbara. They taught me a very simple assembly language. But on the other hand, I've never taken a language course, and I really haven't had time. So I would have really liked to have learned Perl and some of the others, but I've never really had the time to do it. So I really can't answer that. Does that address some of the issues I'm talking about today?

Booch: Absolutely, yes. That very much does. Perl, PHP, Python, Ruby-- those are all the things where people are programming at the edge of the web, if you view the web as the platform. This is an interview about you. We can talk about my opinions about them once we turn off the recording.

Humphrey: Well, that's my loss. I'm sorry I've never had a chance to fiddle with them, and it's probably something that I would have done well to do with the PSP, but I never actually did.

Booch: Well, I think I've covered all the topics but one, and that's just having had such a colorful history and having been involved in so many different things. I'm just curious as to how you might recap your

experiences and ask questions on what you see that has changed vastly in the years of your career, what's stayed the same, and what do you view the future in the computing business to be? And then the last question I'd offer is what advice would you give to someone interested in the computing field?

Humphrey: Well, on the what's changed and what's the same question, that's all about people. I've seen an enormous amount that has, quote, changed. People view it as changed, in terms of languages and the whole structure of our systems and how we design and build things. But quite frankly, it's the same story that I've seen for 50 years. We're basically trying to work with a higher-level constructs-- richer and richer constructs so we can compose our product. We've been on that path all along, and that basically comes back to what I'm saying about languages, that I think we've been down that path without really looking at all sides of it.

So one, I think we've got a whole portion of technology that is, unfortunately, going down a track without thinking about how people will use this stuff. I see, for example, in the whole academic community-- I looked at the computer science curriculum, and to a great extent, the software engineering curriculum. There's very little in there about people. I get a lot of reaction that the most important single thing you can do in producing good programs is to get good people. And I think we somehow view people as once you get them, they're immutable. People are very flexible. I've seen people who weren't performing very well turn into stars. Absolutely extraordinary. With the right motivation and the right guidance, and frequently you've got to push them hard. I've taken people and sat them down and said, "Look. You're either going to start looking for another job, or you're going to have to start doing the following."

Most of them shape up, and it's amazing. We get some really good performers out of people that have those various personal problems and stuff like that. So what I see is an awful lot of that, where our community has gotten so techie, that it really is losing touch with the real problem. The computing problem we all deal with is a human problem. How do we actually take ill-defined problems and reduce them to such a precise format or form so that they can be executed automatically by computers?

So we've got to deal with users; we've got to deal with our co-workers, and involving the people. This is what's sort of frustrating to me about the whole computer science community. Academics don't seem to have any interest in people. They're all focusing on their techie stuff, and I think it's quite limited. It can take you just so far. They've done a lot. I mean, it's good stuff. I'm not against the technology improvements. But I think we've really got to take a much broader view. So that's sort of one thing I've seen that has not changed. It's sort of been steady all along, with a view of this field as a highly techie one without really looking at the human interaction parts of it.

And I think that's why we're having a lot of trouble getting new people into the field. This comes back to your question about what I'd tell people. I see young ladies, for instance, they don't want to go to work in a corner and just hack away. And I don't think they appreciate the degree to which really good programming work is a people business. You're heavily involved with other teams, with users; you're negotiating. The techie part of sitting down, writing code and designing is a really small part of the engineer's job on most TSP teams. There's an awful lot of negotiation and involvement and help and discussion and that sort of thing. So I think that's the issue. I think there's a misunderstanding, and the whole academic community has to much better understand and prepare people for the kind of interactive careers the software business really is.

The Future

So that's where I think we've got to move forward. I think the change side-- certainly an enormous amount has changed. The Internet--I must admit, I was embarrassed to say when I first heard and was beginning to get familiar with the Internet, I remember going to a conference. I was a keynote speaker. It was a UNISYS conference in Tampa here. The engineers were all abuzz with the Internet and how it was going to be used. And I was sort of, "How in the world would you use that?" I had no real concept of where it was going to go, in spite of all of my views at IBM and all that sort of thing.

So that was an enormous change, and I was really quite surprised at how it took off, and it was a real eye-opener. And we've seen that. I mean, the continuing growth of technology, I don't think any of us ever visualized the unbelievable development of the technology in this field. It's continued. It's continued for as long as I've been in the field. That's damn-near 60 years. It's just hard to believe. You literally can't imagine what's going to happen down the road, and I was in the position trying to imagine the future when I was at IBM on corporate staff.

So I think back to what we were thinking about then, and I was thinking about what is an intelligent workstation? And I said it's one that runs MVS. My contention was the minimum computer size ought to be a megabyte. Well, I was so far on the conservative side, even though everybody thought I was a radical. It's kind of amazing. So there's a lot of excitement ahead, and I don't see technology slowing down. I see it continuing to grow.

Advice to Young People

When people come into this field, I think computing is probably the most exciting field you can ever get into. It's an underlying technology to every field we've got in science and technology and for just about everything we do in business and the arts. So I think it's a great place to be. I think increasingly, people in the computing business who are really familiar with it and knowledgeable in it are going to be in a very powerful position for opportunities down the road. But I think that depends on the degree to which they connect that and begin to build people skills in parallel. That's crucial.

There's one other thing I would say here. Maybe it's sort of in closing, and that's sort of a general piece of advice to people, and that is try to keep yourself out of your decisions. I think I mentioned some of the things I got involved in. By and large, as I've looked back in my career, as to the things that I've done and the decisions I've made and the actions I've taken which turned out in retrospect to be mistakes, are almost without exception-- I can't think of any exception, actually-- they were decisions I made that at the back of my mind I was worried about what's in it for me. Am I going to get ahead? Am I going to get visibility? Is somebody going to see me? Will I get that promotion? When that's in the equation, we tend to make very bad decisions. So trying to make objective decisions is extremely important.

I guess I'd say one other thing, and that is luck is extremely important in this business, but I don't believe that unprepared people are lucky. I think when you're prepared and when you know what you're doing and you've got a good foundation, it's amazing how often luck shows up. You get these breaks. Something happens. So we want to count on good luck, but we need to be prepared for bad luck. My experience has been that when you're prepared for things to go badly, they usually don't. So it's been a marvelous career, and I've been very fortunate to be at the right place at the right time for an awful lot of stuff in this business, and it's been a wonderful 60 years.

What's Next

Booch: So what are you going to do in the next 60?

Humphrey: As I think I may have mentioned, I've got three books on the plate, and I'm having fun with playing the piano. The thing I'm really doing, and what I'm really focusing on, is how can I help our team and the people we're working with in this industry really begin to recognize and take advantage of what we've got here, which is why I've got some papers and things I want to write to try to get some of this out there and to get more visibility, or at least more understanding of what the benefits are of what we're doing. And I think we've been able to do a lot of that to date.

I want to put together an article for *Harvard Business Review* I've got planned here. I got a proposal about ready. They're very hard to get into, so I'm hoping I can get them to be willing to take an article. So I'll be trying that to see if we can get that in there, because we need to get a lot of this message out in front of executives. What I'm talking about here is basically getting people and empowering people to manage themselves. It's the whole knowledge-working business that I don't think executives understand. The great difficulty of this-- and it's a real problem we face-- is that you describe briefly what it is you're doing. You try to give a two- or three-minute elevator speech to somebody, and their immediate reaction is, "Oh, we do that already." Just about every executive you talk to, you talk about empowering their people and letting them manage themselves, that's what executives do. "Oh yeah, we do that." And the fact that they really don't is extremely hard to get across.

So getting that out and getting that message understood is a challenge I've struggled with, and whether I can actually do it or not is something I'm going to try to work at, but that's my immediate challenge to try and get that story out. Of course I've another challenge. I'm trying to get better at the piano, which I'm enjoying. That's where we are.

Booch: Very well. So are there any other parting comments, anything we've not covered that you'd like to inject into the record?

Humphrey: I can't think of anything. Barbara? No, she doesn't have any comments. I would like to close by saying I just so enormously appreciate your participation in this, and you've been a great interviewer, and I thoroughly enjoyed our-- I would guess would probably had about 17, 18 hours of interviews.

Booch: Yes, we did. Well, it has been a privilege.

Humphrey: We've got a lot of work ahead to-- I guess they're going to transcribe them, and then we got to edit them. Is that right?

Booch: That's correct. I'm going to stop the recording here. I'll end by saying it's been a great privilege for me, and thank you for doing this.

END OF INTERVIEW