Mandy Chessell CBE

Interviewed by

Richard Sharpe

5th July 2021

Via Zoom

Welcome to the Archives of Information Technology where we capture the past and inspire the future. It is the 5th of July, Monday, 2021. I'm Richard Sharpe, and I've been covering the IT industry from the computing side, since the early 1970s. Today, making her contribution to the archives, is Amanda Chessell, CBE. And if I told you all of the other shower of things that have been given to her as honours, we would have very little time for the interview, because she is tremendously important to the development of the software industry. And I don't mean the front bits, the glitzy bits, where an icon bounces along in, erm, the desktop for Apple—I don't mean that. I mean the software many of you will never have heard of, but it is absolutely vital to the movement and to the proper processing of information is the backbone of our information society.

Mandy, you were born in…in Bath, in 1965, what were your parents doing?

[00:01:11]
My mother was a nurse, and my father was a naval architect, so, he designed, er, warships.

[00:01:18]
And erm, the navy still had a base in Bath, er, it had been taken there during the Second World War and occupied a hotel, did it not, near the Pulteney Bridge?

[00:01:28]
Yes, it did, erm, yes.

[00:01:32]
And you…you were raised in this wonderful aura, of the city of Bath, with all of that wonderful stone around it?

[00:01:38]
Well, for the first couple of years of my life. Because, um, my father's job required us to move every 3 years. So, I've lived in every dockyard town, apart from Rosyth [00:01:51].

[00:01:51]
[Laughs] Erm, and you're, um, so, your father was trained as an architect.

[00:01:58]
Naval architect, yes.

[00:01:59]
Your mother—your mother was trained as a nurse, education was important to them—erm, did they support you in your education a lot?

[00:02:08]
Absolutely, absolutely, I mean, they…they…they, um, bel—I mean, we would, um, m…my sister's and I went to the state school, but they believed that things like travel and, er, sort of wider curricular things are very important, and we had access to a lot of books and that sort of discussion. So, absolutely, and, the fact that my dad was an…an engineer, effectively meant that engineering things were discussed, and he was building and we…we grew up building things, is…is the right thing—from sort of handicrafts all the way through to woodwork and other such things.

[00:02:45]
Did you have brothers and sisters?

[00:02:47]
Two sisters.

[00:02:48]
Two sisters and where are you in that, erm, in that, three?

[00:02:52]
I'm the oldest.

[00:02:52]
You're the oldest, okay, so, you lead the way, excellent, excellent. Erm, how was your, erm, primary school?

[00:02:58]
Er, my primary school was in Gibraltar, erm, you know, dockyard tan--[laughs] and actually, we had a wonderful 3 years there, um, we had, er, the opportunity for, you know, living abroad, er, living and talking to people who don't speak English, I think that is a very important part of growing up. Er, we would visit Morocco, er, because the border with Spain was closed at that time, so, Morocco was our obvious holiday destination. A…and so, I…when I look back on it, I was at the St Christopher's School, which is…was a wonderful place. Er, they…they pushed me… erm, they…they push everybody to be as good as they could and, er, and so, it was…it was…it was a great school and the whole environment and the place to grow up was wonderful.

[00:03:51]
Was it a denomination school, a religious school?

[00:03:53]
No, it was…it was…it was a service school.

[00:03:56]
A service school.

[00:03:56]
For, erm, er, yeah, for British children who were stationed in…in Gibraltar.

[00:04:02]
Oh, St Christopher because St Christopher carried people across rivers, didn't he?

[00:04:07]
He was, he was the patron saint of travel, yes.

[00:04:09]
Yeah, patron saint of travel, excellent. So, you then moved on to another primary school?

[00:04:14]
No, I…I came back…I came back to the UK, I then went to a comprehensive school, which was Warblington School, near Havant.

[00:04:22]
Right. And how did you do there?

[00:04:24]
Reasonably well. The…the way that Havant set up schooling, erm, which I thought was actually very clever, was, er, when the transferred to comprehensives from the sort of combination of the grammar school and the secondary modern, they turned the grammar school into a 6th Form college for all of the other schools, and then the other schools, the…which were the secondary moderns were, erm, then did the schooling from 11 till 16. So, I did my O-Levels at Warblington School and then I went on to Havant College, er, to do my A-Levels, and that meant they could give a wider range of A-Levels, they had specialist teachers for them. So, erm, I…I… you know, that worked very well, and because Warblington was a secondary modern, er, I did woodwork and technical drawing, and…er, er, and…and dressmaking, and all those things were available at the school because it was, er, you know, b…b…because it had all those facilities for, erm, from its past. So, erm, I…I thought that worked very well. Because also, there is a big difference between children who are 11 and children who are—and young adults, who are sort of 18, whereas to actually, co…er, split the ages between, you know, and br…and take out the…the older children I think made a lot of sense. So, er, I felt I was pretty lucky in that respect.

[00:05:38]
Were you sporty?

[00:05:39]

Er, no, I…I…I am—what I've always enjoyed are, erm, the—we used to do outward-type things, so…so Duke of Edinburgh, erm, we did a lot of sailing and, erm, rowing, canoeing, that type of sea--- as I was sea ranger, up in… in Portsmouth, um, and er, so, it was—we did a lot of, yeah—sort of hiking, sailing, that type of…of thing. I…I…I wasn't a good runner, I wasn't, er, you know, the sort of traditional sports day-type things. But those more outdoor, erm, activities, I absolutely loved those [laughs] I still do, actually, I…I've actually just come back from walking the South Downs Way.

[00:06:20]
Well done.

[00:06:21]
Er, so…so, that, erm, that love of the outdoors has, er, has stayed with me.

[00:06:28]
Do you have particular memories of particular teachers?

[00:06:31]
Oooh, that's a good question. Yes, erm, er, I had, er, one teacher, her name was Miss Bowie, she, erm, was the librarian and then she, er, took teacher training and was very interested in history and we did social and economic history of…of, er, Britain. And…and that was fantastic because that went through the revolutions, the Agrarian Revolution, the Industrial Revolution, and then into modern Britain. And that is fantastic to actually understand what is happening to us today because we're going through another revolution where our whole lives are being digitised. I mean, the Covid pandemic has, you know, has accelerated that for…for many people. Erm, and to understand what happens as employment changes, the way we live changes, where we live changes, erm, has actually—was actually really good, erm, in terms of understand what is happening now.

[00:07:24]
So, you see this as a continuum?

[00:07:28]
Yes, yes, actually, and…and…and when you look at the societies, and you think of the Victorian societies, h…how the initial change is—you know, creates quite a lot of hardship, it creates new elites and things. And, then…and then we seem to sort of normalise and…and…and things become fairer and more stable and then we have another revolution and the whole things starts again. So, erm, yeah, I think we're in the turmoil stage at the moment.

[00:07:56]
[laughs], we are indeed. Had you met a computer then in your secondary modern school?

[00:08:01]
No, no, no. The computers were kept in a cupboard and only a very small number of pup…pupils were allowed to access it and so-- And so, I…I did see them in the cupboard, but I never really got to touch them. It wasn't till I got to 6th Form that, er, I studied computer science, so—

[00:08:21]
And what did you do then, when you were, what A-Levels did you do and how did you get hold of a computer in 6th Form?

[00:08:26]
Okay, so, erm, my first year, I…we…you had…you had to do 3 A-Levels and 1 O-Level. So, my O-Level was computer science and then I did physics, maths, and biology. Erm, and then in the second year, I, erm, then opted to do computer science A-Level in one year and I dropped biology. So, I ended up with, erm, computer science, maths, and physics, erm, that's my 3 A-Levels.

[00:08:56]
So, that's a pretty, er, a…a classic, um, erm, combination there which you had on the science side.

[00:09:02]
Yes, exactly.

[00:09:02]
You dropped basically, the art side by then?

[00:09:06]
I did, yes, and…and it's funny because, you know those are things that interest me now as an adult, erm, but I needed those for professional reasons and, you know, I'm learning geology and…and…and…and, er, that sort of thing in my daily…er, you know, sort of—as…as I'm getting older, so, it tends to work quite well.

[00:09:25]
Did you have a view of what you wanted to do when you left the 6th Form?

[00:09:30]
I did, actually [laughs], erm. So, when I was 14, erm, IBM came to my school and we'd had some career advice, erm, but the chap from IBM was the very first person who told me being an adult was going to be fun. Erm, all the other jobs seemed dull and I thought—I got the impression that you really…you might do some training, but you really stopped learning and you just did your job. And the chap from IBM said, the world of technology is constantly changing, you are gonna be c…c…constantly learning new things, it's very exciting. Erm, you're going to see the world change, you're going to be able to travel, you're going to have colleagues and friends all over the world. And at 14, I was like—Yes—[laughs], I love this [laughs]. And it…and…and…and basically, he gave us guidance on what we should be studying and so, you can see why I did the subjects I did and the moment that computer science was offered, I was there. Erm, because that was so inspiring, erm, and I rem—I wish I knew who he was because he set my life on a very good course.

[00:10:37]
Right, let's see if we can find him. What year was that?

[00:10:41]
Oh, my goodness [laughs], I was 14, so, what are we talking about, erm---

[00:10:46]
79.

[00:10:47]
Er, yeah, that sort of--

[00:10:48]
1979, around there, okay.

[00:10:50]
Yeah, that's--

[00:10:50]
And er, and your 6th Form college was in Hampshire?

[00:10:55]
Yes, Havant.

[00:10:56]
What town?

[00:10:57]
Havant.

[00:10:57]
In Havant, in Hampshire?

[00:10:59]
Yes, yeah…yeah, so…so, erm, our next move was to Plymouth, so, I went to Plymouth College.

[00:11:04]
Right.

[00:11:06]
[Laughs], so—

[00:11:07]
Now Hampshire, w…was a big centre for IBM, erm because, in Havant, it had a…a manufacturing plant, making disc drives.

[00:11:14]
That's right.

[00:11:14]
And it also moved its headquarters to Havant, although it kept a headquarters in London and it had a whole new complex headquarters there, did it not?

[00:11:23]
Er, nearby, yes, quite, er, near Cosham, yeah. Erm, so, I actually did, erm, a summer job at Havant plant, er, and then, for my degree, I did my industrial training year in Port…in one of the Portsmouth offices. So, you can see th…erm, that I was recruited at 14 because, erm, you know, my—I was so focussed, so, I did my summer jobs at IBM, I then, erm, did my industrial training in—at IBM, and er, and obviously, they wrote to me when I—and they gave me a bursary for my final year of my degree and wrote to me in that year and said, "Would you like a job?" So, I said, "Yes, please" [laughs], so, I was very, very focussed and I've never worked for any other company, erm, from that day. Erm, so, they did a good job, you know, from in terms of recruiting, they did a good job.

[00:12:16]
Very Jesuitical, they got you young.

[00:12:18]
They did, yes, yes—

[00:12:20]
Which is fine…which is fine.

[00:12:20]
But you know, as I say, it gave me direction, it gave me, erm, an insight into an industry that the school knew nothing about. I mean, I would not even have thought of this career, if he hadn't come. Er, and I think that is an incredible lesson for all our engineering disciplines that school children are not inspired to do engineering 'cause they don't know what it means, they think of it as fixing washing machines and they don't appreciate the—how interesting and the people—the wonderful people you work with and, erm, how important what we're doing is. That's… that I…that I find, erm, you know—I think it's er, it's something that…that we really have to think about, erm. Because most teachers, they—how…how could they know, how could they know what we do?

[00:13:09]
And you've still got the buzz, haven't you?

[00:13:11]
Mmm, [laughs] yes, yes, I have.

[00:13:15]
Well, there's nothing wrong with that, I think that's…that's fabulous, you've still got the buzz after, what 35 or so years in IBM.

[00:13:21]
35, yeah-- 35 years at IBM and, erm, but I've changed jobs every 4 or 5 years within IBM, as—and I've worked on every wave of technology, er, and typically, I work on what we're going to be selling in 2 or 3 years' time. So, I…I…I typically do tho…the sort of the first stage implementation of new products, new ideas. And I work with companies that are plan—that either head in their industry or they're planning to make

a…m…m…make a disruptive change. And I work with them on what technology do we have today that will help, but also, what are they going to need? Because when you think about those leaders, they…they're going to force the rest of the industry to change and that change will then mean if we've got products in place that support that type of business, then, of course, then, that's…that's our next generation of…of…of things. And that's…that's what I've done, er—my very first job, I was working on mainframe coding, a very, erm, a…a…a very valuable but somewhat old product, which is still running today and still doing incredibly well, this was the CICS-- CICS, er, so, it's transaction—mainframe transaction processing system. And then, erm, er, and then after that…that, which was a really good training, we can go into a bit more detail.

I then spent the rest of my career on—[laughs] I did the first, er, transaction processing work on UNIX boxes, er, because, at that time, no one believed you could use UNIX for business, you had to have a mainframe 'cause that—you know, so, it's completely ridiculous what we were doing. Of course, that's now a mainstay for the business. I then did a lot of the early object-oriented distributed object for the COBOL work which led into J2E, which is, er, the sort of object transaction processing. Then into work on Workflow, and various other things. And as I say, we can go through this in more detail.

[00:15:26]
We will.

[00:15:26]
But what I was trying to show you is the nature of the work that I've done and so, even though I've been in the industry a long time, erm, I am doing the ground-breaking work that will be, you know, t…the standard part of the business, going…going forward.

[00:15:42]
We're going to go—we're going to run that reel back in a moment.

[00:15:46]
Yes, fair enough.

[00:15:46]
But I want 2—I want 2 snapshots from you, please. The first is, snapshot, number 1, what was IBM like when you joined it in the mid-80s?

[00:15:56]
Okay. Er, so, this was after I'd graduated, and erm, we were working—I was working—I moved to Hursley Park, which was our research and development. And at that time, erm, there was a lot of concern about losing our intellectual property, so, it was very com…compartmentalised, you sat in your department, you went to…to lunch with your department, um, so, it was—you were—we were in lots of little self-contained bubbles within…within the laboratory and, erm, er, there was a very strong focus on quality, erm, we were very…very strongly trained in terms of software engineering, erm, and—but…but you know, you were…you…you were in this bubble, your view of IBM came only from your manager—that…that's…that's the…that's the way it was then. Erm, so, that's what—snapshot 1.

[00:16:53]
It was a highly pa…erm, paternalistic organisation?

[00:16:56]
Very.

[00:16:57]
Where…where your career was being looked after, was it not?

[00:17:00]
Yes, by your manager. And…and…and all career options and what was going on in the company came through your manager, erm.

[00:17:07]

Right, okay, erm, and that began to be shaken about and wasn't flexible enough to deal with the new world, and IBM lost its position, lost its, erm, chief executive off…officer, who had been internal—

[00:17:23]
John Akers, yeah.

[00:17:23]
And then in…in came a man called Gerstner, who threw the whole thing up in the air, and he claimed that this elephant of IBM could dance.

[00:17:32]
Yes, and he was right.

[00:17:34]
You could feel that impact?

[00:17:36]
Absolutely. I'll tell you, the ver…I mean, I'm very junior at this point. Erm, I know the company is in trouble, erm, you know, and it's like oh, my goodness. And I come in and I'm the first one in and I open my email and there's an email from… from…from Lou Gerstner and it's like, oh, my goodness, what have I done? [laughs] And I…and I read it through, and I thought that's very interesting, why has he sent me an email? [laughs] 'cause he was the very first CEO to email all the employees and what I hadn't realised was everybody got a copy of this email. But he…he…he explained and…and his book is amazing because it's got a reprint of these emails and I remember getting them, erm, and actually having a conversation with him. And he did a number of things, he created the DE's, the Dis… Distinguished Engineers, which is the job I have today. Erm, he created the IBM academy of technology, which created, erm, a sort of horizontal organisation of all the technical executives that—well, not all of them, but…but the majority. And that…w…we—that was how we did a lot of the cross-organisational, erm, technical work, particularly the advance work that I…that I was a part of, and that was where I started to build that global

network that was promised to me when I was 14, [laughs], so, er, yes, so, I…I…I…I thought he was a very great man, and he…he made some very bold decisions, erm—

[00:18:58]
He was…he was a great communicator, wasn't he?

[00:19:00]
Yes, he was, yeah.

[00:19:02]
I believe, now, I'm not sure this is right, and I'm testing this with you, one of the decisions he made, was to scrap a massive project called Systems Application Architecture.

[00:19:13]
Yes.

[00:19:14]
--erm, to…by IBM to solve what was then considered to be the programming or coding crisis.

[00:19:21]
Yes, it was, and…and…and—

[00:19:22]
Did he scrap it?

[00:19:23]
Sorry?

[00:19:24]
Did he scrap it?

[00:19:25]

He did, yes, yes and…and I was actually working on one of the…the API's, um, and…and the…the big floor, the idea was right, and we have been trying to do it as an industry ever since in terms of staff—

[00:19:37]
What was the idea?

[00:19:37]
The idea was that rather than having a different programming language for every product, you would start to say, well, let's say, there's a pr…programming language we need for managing files, and there's one we need for managing databases, and there's one we need for managing communications between programmes. So, let's have standard interfaces and get all our products to implement them, then customers can plug and play between their different products. So, that sounds, you know, we…we…talk about this today, we've done web services, we've done microservices, you know, we do REST APIs, we do this all the time. The problem with…with the SAA was it was just IBM, and if we continued to dominate the industry—'cause, when we were so dominant, we always had, products competing—IBM products—we had 2—at least 2 of every type so that there was competition because we didn't have external competition. And that was one of the things that…that…that threw the company because we were set up to compete internally. And then when the c…the external competitors came along, of course, we weren't watching 'cause we were used to competing internally and that was a big cultural change, to go to having—to thinking externally, so, we had our own networking system called, er, SNA and we then moved to TCP/IP and all the open standards for things. And for a long time, we supported both but now, of course, that is the industry standards is…is dominant. And this has been…a th…a theme throughout my career is that we might do something proprietary to start with, but really, what takes off is the open standards and the use of open source. Er, that's…you know, that's the thing that allows everybody to play and that's when you get growth, erm, in terms of, you know, use and, er, and economic value.

[00:21:30]
Was that, Mandy, a hard battle to win, with inside IBM?

[00:21:34]
Er, yes. Erm, you know you always, erm, I mean, I…I…I worked…I started—One of my very early jobs, I switched to open standards and…and things and I'd been called a traitor and…and things like that for—you know, then…the…this is a standard innovations dilemma, right, you have a successful product, here is the next generation, and in some respects, as you work on that next generation, you start to damage the income of the old product. So, um, you know, it's…it…it…it's…it's the cycle, right, that's…that's…that's the way… that's the way of life in IT, is, that, you know, that each generation is slightly different, there are a lot of similarities and as you move through your career, you pick up the next generation faster than a lot of people who are new, because you've seen it before, right, and all you're doing is looking for the differences between what we had before and what we had now.

Erm, but it's always a move to open—open and the open-source movement underneath it, which is creating those, um, sort of de facto standard implementations of technology. I mean the internet is almost all open source now, isn't it, there…there's very little, er, and think how successful that's been, erm.

[00:22:50]
Yes, I wish I had an open-source operating system on my PC here which—

[00:22:53]
[Laughs]

[00:22:54]
But I got Windows and it crashed, which delayed our interview. I'd much prefer to have that wonderful operating system called Ubuntu, which I'm sure you…you've heard of.

[00:23:02]
Yeah, Linux, yeah.

[00:23:03]
It's rather good—rather good. Erm, so, basically, here you are, starting in IBM, it's…it's paternalist, it's relatively closed a…as a community—as a culture. You're even fragmented within that culture. Your career is looked after quite considerably, whereas you would describe today, IBM is much more outward-looking, er, much more open, in both sense of open source and open to new ideas. And of course, much, much, much more based on software and services than on hardware as it was.

[00:23:37]
Yes, and…and…and the openness is—and we have an internal intranet effectively, it's called W3, and everything is shared and that's…that's the thing that…that…that really opens up, IBM-ers eyes to the wide…wider world. Because I can look up and find out about how a certain thing happens in a particular industry, erm, we can…I can look at things that research teams are doing, er, so there is a l…lot of stuff around content computing, er, I, er, you know, I can—you know, I…there's information about different customers. So, we have a lot of, so, erm, sort of…eh, eh, sort of, almost internal social media around what we're doing and that's…that's…that's lovely and i…it again, if people who are joining, they get…they get to see, erm, the bigger picture. And it can be overwhelming, [laughs] it's like—where do I start? Er, but also, it…it does give that, um, that…that breadth and depth which is just lovely.

[00:24:35]
Two years before you were born, the… then IBM, erm, then they take a massive gamble and launched the 360 range of computers, and including, getting an operating system eventually going. Erm, and er, the year you were born, there were 29,600 computers in the world, would you believe?

[00:24:57]
Mm-hmm.

[00:24:56]
29,600. And also, the year you were born, [coughs], a man called Gordon Moore said, "Hold on a minute, I think we're doubling the number of semiconductors that we can put in a space every 18 minutes—18 months"—not minutes—18 months. So, you

were really, um, there and…and with these large mainframes, they were really designed for batch machines, weren't they?

[00:25:21]
Mm-hmm, yes. We…we did have, we did have some interaction, um, which was, sort of, page by page, er, but largely and, I mean, certainly initially, er, it was…it was batch, it was back office-type admin, administrations—

[00:25:36]
You run the payroll?

[00:25:37]
Mm-hmm, yeah, payroll, um, the accounts, all that sort of thing, yes, yeah.

[00:25:44]
And then somebody, somewhere, er, er I think it was a utility, was it not, in the United States said, "Hold on a minute, we want to be able to access information in a transactional way."

[00:25:54]
Mm-hmm.

[00:25:54]
And therefore, we need to do this. And other said, okay, we'll write it into each programme and each programme—

[00:26:01]
Yeah, nightmare [laughs].

[00:26:01]
And each programme will have its own… own little transaction. And then someone else in IBM, or it might have been a customer I think, in a utility, said, "No, no, no, no, let's have a common interface for all of these transactions" and therefore, they

produced this product—still running, still vital in my opinion, not talked about enough—CICS, called---

[00:26:24]
Yes, yes, yes, that's right, yes.

[00:26:24]
Customer Information Controls System. And that was your—to continue developing that, that was your first, main job in IBM, was it not?

[00:26:34]
Yes, I worked on it—

[00:26:36]
You were a developer?

[00:26:35]
Yeah, I worked on the…the As-assembler code, er, from those original machines, erm, and I worked on what was called terminal control and that was how each CICS system talked to other CICS systems. And er, so that was, yeah, that was very interesting, it's…it's 'cause you…you had to…the…the way programme worked in those days was the…the memory was very expensive. So, depending on whichever programme was, in…in charge, we used to sort of copy the memory over…over and so, you had to really understand the whole system because when it failed, it would give a dump of the memory, and you needed to know what running in order to know what—be able to interpret what the fields were. So, er, it was—and it was very, very detailed and all the variables were, like, 8 characters long, so, erm, they were all…they were all sort of erm, er, acronyms and things. So, each control block which was sort of an area of shared memory, er, would sort of have its own sort of 3 or 4, erm, digit common thing, and then you'd have 4 letters to actually describe the variables. Erm, and often…often you only had 3 and I used to play games when I was driving into work, was—for every number plate that I passed, I could tell you what the variable name was—

[Laughter]

And when…and when I could—you know, er, day by day, I could do every single car that went passed, I realised maybe it's time for a new job [laughs].

[00:28:02]
A true techy, that is a wonderful story. So, you were erm, a developer on CICS/ESA?

[00:28:09]
Yes.

[00:28:10]
Where would you place yourself in the hierarchy of… of IBM, in Hursley laboratory then?

[00:28:15]
[Laughs], oh, a…a…a…a…a…a mote of dust, I would say [laughs]. So, um, I, yes, I was very, very junior, erm, but I ended up, er, because I was good at this detective work in terms of understanding so we could make a TOKA system, they could fill it out with stuff and crash it, um, and, uh, I could tell, yeah, yeah, I got very good at working out what was actually running at the time, what caused the problem and, er, so, I got very good—I…I spent 4 and a half years working on that, um, and I did get very good at it, but, as…as you say, after a while, it doesn't—it's not good for your head [laughs] 'cause you're…you're—you're in so much detail, particularly with the assembler code, um and at that time obvious…IBM was not doing very well and I was thinking, all I know, is IBM As-assembler, erm, and I know the internals of a CICS system, I don't know how to run it—it's a very different skill to working on inside something than actually using it. So, um, I was worried about my skills and, erm, we were just starting a new…new, erm, project to put CICS onto Unix. And I said, "I want to go and do that." And they, erm, my boss said, "Well, we need you to continue doing this so, let's wait a year and we'll train someone else up, and then you can move." So, I said, fine. And this was very naïve of me, erm, because, of course, they didn't train anybody, and a year went by and I said, "Right, I want to move now." And he was kind of—"Well, we've not trained anyone you really can't move."

[laughs] and I just went ballistic, I said, I—you know, "I really need to change, I find it very difficult even coming into the office because I've done this job for so long and I need something new" and I was moved, and I think he was so shocked, er, I was actually moved the following week.

And I w…was moved into the test organisation of the new product and it was supposed—I think it was supposed to be a punishment, but it was excellent because going from a mainframe environment, which is so different from Unix, erm, I needed to—you know, I couldn't even use the—I couldn't even log on [laughs]. So, the testing was—I did 8—18 months of test, um and I tested a lot of things. There was a thing called, a distributed computing environment, DCE, which was brand-new, in terms of thing—providing something called remote procedure call, which—and so it had a directory of all the programmes and their functions, and you could start calling things. Because we were now going into a distributed world.

[00:30:58]
Right.

[00:30:59]
And focusing on distributed computing, erm, I had to learn C, I had to, erm, to learn the editor, like, VI, which is the editor at the time. Erm, and so, it was a completely new world, it was fantastic and 18 months in tests was like, a really great thing. And then, erm, and then I moved onto the CICS product on AIX, which was IBM's Unix.

[00:31:23]
Yeah, okay. Two things I want to pick up here. One, you're quite assertive when you need to be, aren't you?

[00:31:29]
I'm quite assertive--?

[00:31:31]
When you need to be?
[00:31:32]

Oh, yes, yes, it's…it's erm—yes, it's difficult because, I can't shout, erm, you know, I can't get angry because that's not what women do—if…if you…if…if…the…if a man gets angry, he's being masterful and forceful but if a woman gets angry, she's—I'm not going to say the word, but it's not polite. Erm, so, I can't get angry, I can't shout, erm, so, it's how do you…how…how do you, er, assert yourself [laughs] and still be ladylike? And that's…that's actually one of the…the challenges that we have. And…and a lot of, erm, things, er, things around Zoom, are…are actually helpful to women because you get a fair—it's a lot fairer [laughs] in terms of, er, erm, time and you know, sort of, t…time and opportunity to be heard. And…and this is it's a much more of a leveller, which is, you know, surprising.

[00:32:39]
So, a whole new technical field of software opened up in front of you and in the middle of it, was this…is this, well, I won't describe what I think of it, but I think my tone of voice will tell you—Is this programming language, [laughs] called C—what do you think of it?

[00:32:55]
Yeah, I loved it, yeah.

[00:32:56]
Did you, why?

[00:32:57]
Well, hey, I'd come from the Assembler—

[00:33:00]
[Laughs]

[00:33:02]
Erm, and er, a…a…and the thing about C—and the…the and the most important thing with any language is its symmetry. Er, and I was used to manage memory—I mean, in the As-assembler, you manage your own stack, so, [laughs] having to manage—I always manage memory, uh, you know, it's, er, so…so C, for me was, erm, it allowed

me to think in bigger blocks, I could do bigger programmes because of the thing. So, as far as I was concerned, that was a major step forward, and we were—'cause we were—we were heavily criticised for using C because it was considered to be, you know, sort of, not properly technical, er, the compilers were not as efficient as the As-assembler, as the As-assembler programme I could do. However, within 18 months of us using it, the C compilers were faster than anything that a human could produce. Because they were doing types of optimisation that you—were not economically viable for a human to do. So, the…it was exactly the right decision, but, it was, as I say, initially, the fact that we weren't going to As-assembler, er, was considered completely ridiculous. Erm, but, you know, this is…this is the world and Moore's Law, of course, is one of the reasons why we got away with it. Erm, because there was plenty of processing power to allow the optimisers to run, erm, on our, on our code.

[00:34:28]
And all of this time, you're working at Hursley Park.

[00:34:31]
Yes.

[00:34:32]
Did Hursley, erm, am I right, it's Jacobean, it's a Jacobean house?

[00:34:39]
Wooo, I don't think so. I think of it, erm, it may, yeah, erm, it does have—it has 2—it was built in 2 phases. Erm, and there was an Elizabethan—

[00:34:50]
There is an Elizabethan part? Okay.

[00:34:52]
Yeah, and the—the other, well, that…that…that was destroyed, and in fact, you can see—in…in the summer, you can see the old foundations of the Elizabethan house.
[00:35:00]

Right.

[00:35:01]
And then, the more modern—it…it…it's a…a combination of brick and marble, erm, and it was…the original house was built and then, erm, the, er, w…whoever was, you know, lord of the manor, or whoever he was, er, he married and met a rich American heiress and then the wings were built in the second phase. Er, that's my potted history, I don't—

[00:35:25]
Yeah, that's fine. And it's in leafy Hampshire, er, and it's in a…a wonderful location, but it does have some rather awful modern buildings around it, does it not?

[00:35:34]
Erm, I wouldn't say they were awful; they are hidden by trees—

[00:35:38]
That's probably the best-- the best place for them.

[00:35:40]
[laughs], yeah, they…they provide air conditioning. I wouldn't say they were…they were particularly horrible. We did have some '60s buildings and they've all been demolished.

[00:35:48]
Aha, that's what I'm thinking of.

[00:35:49]
Okay, yes, no, they've gone, erm, and so, now everything is red brick and white to match—We have, erm, a sort of very glass building, which is hidden off the back. But no, I would say, erm, it's a nice combination because we have the old house—we use that for the customers, erm, but the…the modern part, erm, where er, you know, when we're working does have air conditioning, it does have, er [laughs], erm, it's

bright and light and, er, it's designed to…to…to create a good working environment for…for programmers.

[00:36:24]
IBM was going through phases of becoming an international and then a multinational company. And in 1974, it made the, erm, very large—for your career, important decision of moving development and support for CICX, to Hursley.

[00:36:41]
Yes.

[00:36:42]
Erm, which was a very, erm, big step for it, it allows one of its crown jewels to go outside the United States, and that's what made Hursley actually so important. Now, by 1981, CICS had 16,550 licences, would you believe?

[00:37:02]
Yes.

[00:37:02]
And each of them big licences, we're not talking about, you know, 20 dollars here or there, these are big systems, doing very important things. Making sure that when you take your money out of your ATM, the whole transaction works from end to end properly, instead of falling down somewhere. But underneath this, as you have so, erm, adequately described so far, is another type of approach to computing. Which is, oh, let's have a little mini-computer or something like that, with a rather rough and ready operating system, let's call it Unix, running on it, and we could have one, say, in each department, just as they have one in engineering, we'll have one in our department. So, this led to a…a huge distribution of computing and a great threat to the wonderful citadel of the IBM blue mainframe.

[00:37:52]
Yeah.

[00:37:52]
IBM has to respond to this, what is the response?

[00:37:56]
Er, to do Unix as well? That was…that was it…it's…it's like, okay, there are going to be machines around it, we need to make some of those, erm, IBM machines, er, we need to—allow our work to be distributed onto those machines. And the reality is, you know, the mainframe is still there, it's still doing…doing it…it's…its thing that the growth has been in the…in the…in the distributed systems.

[00:38:25]
Because IBM still has a mainframe operating system called z/OS, does it not?

[00:38:25]
Yes, and it's pretty much the same one as we had before, you can still run programmes, that were written for system [coughs] 360, erm, they will still run on the…on…on…on…on…on Z, or [laughs] as you may want to call it, system Z.

[00:38:42]
Z, yeah, sorry.

[00:38:43]
Yeah, [laughs].

[00:38:45]
Erm, so, IBM makes this crucial decision of taking one of its top…top proprietary pieces of…of software, of plumbing, CICX, and it's going to put it into the open systems environment and make it work on open systems, and you begin to lead that project, do you not?

[00:39:04]
Yeah, so, even though—I was still very junior, erm, but because, erm, not many senior people were prepared to destroy their career by moving onto these little machines, [laughs] er, there was, er, er-- without knowing our…our…our, erm, sort of

promotions scheme, I was, erm, what was called a—I think it was an advisory software engineer, which meant I was officially managing, sort of leading 2 or 3 people—a small team, that would be what I would be expected to do. And actually, there were 4 of us, we were the mo…4 of us were the most senior in the product. Erm, and we each had a quarter of the product, and we did everything. We did the planning, we did the design, we led the programming teams, we spoke to customers, we helped, erm, you know, with servicing the product, if there were problems with it. We wrote to documentation, erm, and so, for me, at such a junior level, to have such a broad responsibility across an entire product, you think about, what I learned doing that. Erm, and we were just really lucky because the…the 4 of us, you know—we…we normally, you know you might have had, normally you would have more senior people above you who are making those sort of decisions, but we…we did…we did all ourselves. And—

[00:40:23]
You were in your late 20s?

[00:40:26]
Sorry?

[00:40:27]
You were in your late 20s?

[00:40:26]
I was, yeah, so, let me think, if I joined at 22, I did 4 and a half years, er, which would bring to me to, er 26½ so, yeah, late 20s—late 20s.

[00:40:39]
Amazing.

[00:40:40]
Mm-hmm.

[00:40:41]
Were there people inside IBM who said, this will not work?

[00:40:45]
Absolutely, yeah, no you can't… you can't possibly run a business off of a mainframe, you have to use a mainframe because it's the only thing that will guarantee the transaction. And we knew we were guaranteeing transactions, erm, and er, but, you know, it was—that was the belief. And the…and the thing about the…the…the mainframe systems, you run them for a couple of years, and they don't fall over. Unix machines, of course, we were booting them every, you know, few weeks or whatever, and that was considered, well, this is why you can't run a business on it. But, of course, if you're transactionally secured, if you're backing it up and things like that, you can reboot it, erm, and, you know, we…we run our business on machines that are being rebooted all the time. Erm, and it's…it's a different mindset, erm, on a mainframe, it should never fail, er, and the distributed systems, are designed to be restarted and in fact, we use a lot of replication, to allow rolling updates, to…to… through…through the systems. So, different…different technique.

[00:41:43]
And different techniques are called for you because—from you, because, of course, you know your technology and you're excited by it and excited by the newer technologies, but now, you're a manager.

[00:41:56]
No.

[00:41:57]
Not yet?

[00:41:58]
No, not—

[00:42:00]
But you're a technical lead?

[00:42:02]
I'm a technical lead. Now, this is very important, because IBM, has what we call a dual career path.

[00:42:08]
Right.

[00:42:09]
And what they observed was, the last thing they want are their top engineers wasting time doing admin. So, the technical career path-- so, I'm, um, a technical executive, a Distinguished Engineer, and the—you can be technical and not a manager, all the way up to executive and then you…you work with—typically, you work with a manager who is one level above you. But I mean, so, we have technical people right the way up to the most senior teams in…in the company. And that means, that there are people who can speak executive and can speak technical. Because think about where the new ideas come from—they come from the…the…the people who are with customers, the people who are coming into the company. And so, you need, er, technically eloquent people to work across it. So, I was not—the first time I became a manager—I'm trying to think, it…it…it was—I've always had—for my level, I h…have a very small team and it's just basically a group of people who are supporting me in my technical executive role, er, because most of my colleagues are sort of…sort of peers, are in the US. Erm, and so, I have a small team in the UK, and they support me on my…my innovative projects. Erm, so, r…r…really, i haven't done, you know, the sort of director of IT-type role at all. Er, but what I am…what I am responsible for is the development and the insight for what are our next generation—what's…what's the next disruptive change?

[00:43:53]
So, CICS on open systems was launched, and…and erm, did it make money for IBM?

[00:43:59]
Er…eventually, but er, not initially, er, because—and this is the thing with all, er, middleware, it takes about 5 years to get the product to a point where it…it…it…it, you know, you can run business on it. Er, and…and that, of course, is very expensive.

So, the payback is then 5 years plus. Erm, so, um, our m… middleware software that has been around a long time, is very, very profitable, but it's a slow—it's a slow…it's a slow give—erm, sort of payback on the technology. So, yes, it did eventually, but erm, it took a long time.

[00:44:43]
Which must be frightening for a company?

[00:44:45]
Well, not one as big as IBM, erm, we always worked—always had been very clear, this is a Horizon 1 technology, this is Horizon 2 technology, this is Horizon 3, Horizon 3, massive focus on customer, erm, satisfaction because this is the money engine. Horizon 2, we're always looking at h…h…how, erm, how do increase our market share 'cause this…this is the growth, this is going to be the engine that's going to…to pay the salaries, erm, you know, going…going forward. And then we have the Horizon 1, C's-- where we're working with these specialist customers, erm, building that and proving…proving the next generation and improving your ideas. And…and when you have that very clear distinction, erm, people know what they're trying to achieve, and er, and…and…and you've obviously, ha…ha…then have a business that is going to go on, [laughs] and, er, you know, the company has been around a very long time because of that constant recognition, that you have to have your…your different hor…you're different waves of…of technology—make sense?

[00:45:48]
Erm, of course, it's a highly technical area—software, but as well in my—looking at it, and, also, being involved in software development, erm, for some years, before I became a journalist and researcher. Erm, I do think, and maybe you're going to disagree with me, which would be fine—I'd really like if you did. I do think it is quite often an area of fads and fashions.

[00:46:13]
Mmm. Actually, I'm not going to disagree with you.

[00:46:16]

You're not?

[00:46:17]
Um, because it's…it…it…it's a frustrating, it…it…it…it's…it's frustrating that, erm, because often what drives the fad and the fashion is—oh, that's complicated, let's make it simpler. And each generation starts with a very naïve programming model then…then everything, you know, it doesn't have security it, it doesn't think about upgrade versioning, all those things that we know we need, um, and everyone thought, oh this is so much better because it's simpler. Erm, and then, of course, we have to bolt everything else on eventually and it becomes exactly the same as the previous thing. And so, some of this is reinvention for its own sake, some—often each generation does give you something slightly better, erm, but erm, there is one area that is particularly bad at the moment that we're suffering from, is…is the user interface. Erm, because there are so many frameworks, and the frameworks have a fashion for 5 years and then there's a new framework and you have to rewrite everything. So, yes, we have a lot of fads and fashions, erm, we are moving forward as an industry, and in an incredibly fast way if you loo…compare us to other industries. Erm, however, we do waste…we do waste, erm, a lot of resource on—redoing things with the new spelling.

[00:47:42]
Since I've been around covering this…RAD came along, Rapid Application Development, now, that's gonna…that's gonna solve it, forget all of this cascade stuff, that's old-hat, we can get stuff out the door really quickly. Oh, goodness me, it's just fallen over, what do we do now?

[00:47:58]
Mm-hmm.

[00:47:59]
Object orientation—oh, let's build it as objects. And then, the super-muscular objects came along called agents, let's build it as agency, what's your reaction to the last two—objects and agencies?

[00:48:12]
So, erm, you know, these programming models are a way of looking at the world and w…what we're doing when we're building software, is, we're creating a digital version of the world and there are some problems… and almost every problem, you can sort of say, well, it's—objects work very nicely in that case. So, the nice thing is you can compartmentalise the code and have it each focusing on different areas, but not—the whole world isn't like that and the trouble with object pro…ordinary to programming is the data gets trapped in little pieces inside the object. And, generally, data is about bigger things and is used in lots of places, so, you end up with duplication across it.

So, there is beauty in each of these styles—none of them work for every aspect of business. And so, the real trick is, how do you know when-- let's…we…we need object-orientated for this bit, we need functional for this bit. Services work here, etc., etc. And it's how we blend and how we— 'cause…cause, object-oriented design is, as I say, it works really well for so many things, but there are other areas and when it's…it's not d…d…done with good engineering design. Because at the end of the day, there seems to be a dream of being able to create software programming language that doesn't require an engineer to actually design it, and that's never been the case. There's a…a… you know, it's…it's talked about, you know, if you remember the…the purpose for COBOL, was that managers would be able to read the code because it had English in it [laughs], and of course, that's not true. Erm, and it still needs designing. The…the whole process of, take…building the abstraction, actually looking at the world and saying, you know, we need to…we need a control system for the electrical power grid, and to actually be able to abstract what is important, what do we need to capture, what can we capture and how do we control it? That still…the software skill—the software engineering skill that never…never goes away whatever style we use.

[00:50:18]
You talked a process there, erm, where are you on the great debate about formal methods or not formal methods?

[00:50:27]
Erm, a…apply it where you need it. So, erm, you know, if you're designing—what you get with formal methods is—it…it gives you a way of looking at the system in different ways. And that's…that's actually key to quality, is not just to do it in one way, but to say, okay, now, let's step back and look at it from this perspective, look at it from this perspective. And so, there are aspects of formal methods—I have used formal methods for some sort of very mission-critical-type software. Erm, and it…it…it helps you pick out the gaps nice and early. Erm, but there are other software where it's completely over the top and in fact, can take you into too much detail too quickly. Erm, and so, you don't get the structural architecture correct because of the…the formal methods. So, erm, I…I…I'm afraid, I'm a…I'm a—use the right thing in the right place type person [laughs]. Er, and er, erm, yeah, so, I…I don't think, I don't think there's any problem that you could say, this one approach is always the best and you know, you never… you never want to move from it. You know, I…I've used all and I, erm, er, and I see there's places where absolutely you use it—you use the more formal approaches and there are other cases where it's not economically viable.

[00:51:43]
So, you're not really an ideologue, apart from being a utilitarian?

[00:51:48]
[Laughs], I'm not sure I know these words.

[00:51:51]
Well, you don't have an ideology which said, we must do it this way, but you—

[00:51:54]
No.

[00:51:55]
But you said—look at utility.

[00:51:57]
Yes, absolutely, absolutely, yes, yes, I'm an engineer basically, which is about making things work practically.

[00:52:04]
March 1997 [cough] became, erm, lead architect of the component broker.

[00:52:11]
Mm-hmm.

[00:52:13]
C'est ça, what is the component, what was the component?

[00:52:15]
So…so, this was very important, so, there had been a whole set of standards created called CORBA, the Common Object Request Broker—I think is its proper name. Erm, and these were, er, allowing us to write objection-oriented programmes but each object could talk to an object in a different programme. So, it's a distributed architecture, there was a thing called the ORB, which was the…the broker. Erm, and what we were doing was saying, well, can we build a m…middleware system that's transactional where, erm, you've got lots—because we—because at this time, we were already realising that everything was now, broken down into lots of little bits and distributed but businesses needed to work coherently, so, how do we start to bring it together?

So, this was, er, the first commercial implementation of this open standard, again, it was like—this will never work, you can't do this, it's—you know, erm, and we, erm, we're using a new language called C plus, plus, er which added object onto constructsy. Er, so that was a fairly easy transition, erm, and erm, at the time, I did a—I started a modular master's degree and learned object-oriented programming through that. So, doing a master's degree sort of, almost 10 years after starting, after graduating, was actually very good for…for this industry because I learned a whole new set of techniques that w…w…were not even thought about when I was doing my degree. Er, so, I—

[00:53:42]
Whe…where did you do that?

[00:53:44]
That was at, erm, well, it was…it was…it was… it was, er, erm, a consortium of universities, I think it's called the Jupiter Consortium, and each…each university offered a master's module for—that they…you know that they were particularly expert at. Erm, and there were lots of universities that signed up for it, but what happened was that there weren't enough students for the capacity that the universities opened up. And so, we…it had a…a rocky couple of years where, um, they would get like…like—get 2 or 3 students for each time they run the course, which wasn't viable. Erm, and so, er, a couple of, erm, er, universities, so, like Brighton and Hatfield I think, erm, basically said, well, we're going to guarantee—we're not going to can—because they were all getting cancelled.

[00:54:31]
Oh, right.

[00:54:32]
Erm, we're go…every module we offer, we're going to guarantee, so, all the students just piled straight into it and so I ended up doing the majority of my modules at Brighton. And that gave…and…and…and that worked…that worked really nicely, and it meant I could study. I would do like 1 or 2 modules—you had to do 10, I did 1 or 2 modules a year. So, it was the slowest master's degree you could possibly do. Erm, but I… but I was working at the same time and that was r…a very good way of feeding in new ideas into the work that I was doing.

[00:55:07]
Let me ask you that question again, did, um, component broker, make money for IBM?

[00:55:16]
Erm, not initially [laughs] so—this…this is always the way with these…these innovative projects. So, we wrote C plus, plus. At the same time, there was a set of—

there was a new language called Java, and there was a set of very similar standards, being developed for Java-based component objects. And there was about to be a battle and one of our IBM fellows, Tony Storey, I think he's in the archive as well. He, er—

[00:55:43]
He's about to be.

[00:55:46]
Oh, okay, good. Well, he—er, you should…should ask him about this—

[00:55:49]
I think I'm going to interview him.

[00:55:51]
[Laughs] he…he, erm, he basically said, "We cannot have this war, we have to bring the Java standards and the CORBA standards together." And that became a thing called—well, initially, it was called J2EE and then it…I think it's called, er, JEE now, but these are the enterprise object standards, er, for Java. And so, actually, we decided to stop working on component broker, which was a C plus, plus run time and we built a thing called WebSphere. And WebSphere became incredibly profitable, er, we had WebSphere for the mainframe, WebSphere for the dist…distributive process, erm, erm, erm, pro—erm, platforms, particularly Unix. And that became the basis of a huge software empire for IBM. So, as I say, component broker itself, er, no, because it was overtaken by events, but what we learned, we actually designed the idea of a container, erm, er, my team did the first transaction—the first implementation—commercial implementation of the transaction service from CORBA, that also was embedded into WebSphere. So, it…it was a massive revolution in terms of how distributed systems communicate and are broken down and…and described.

[00:57:09]
You went on to be, um…um, the project lead for Business Process Beans—WebSphere.

[00:57:17]
Mm-hmm—yes.

[00:57:19]
And this is leading up to, erm, a seminal—

[00:57:22]
But there is a story—yeah, there's a story behind that too.

[00:57:24]
Tell me.

[00:57:26]
Yes, so…so at the time, Workflow systems were mainframe oriented, and what Business Process Beans were, were a Workflow-type capability, erm, but it was also message-driven, programming language to extend Java, the…the Java enterprise beans, erm, and er, I worked on it. We did…we did a lot of prototypes, we worked with a large bank in the US, erm, and er, got it…got it going. Erm, but then, politically, Workflow is owned by our German lab, and so, the work was moved to Germany. So, it was a…a really, erm, amazing innovation time, but ultimately, the work didn't stay in Hursley, it was moved to Germany and then they went and said, "Well, this is how we build Workflow" so, it didn't actually get anywhere. Erm, but, you know, from a sort of learning point of view, and actually exploring the intersection between message-oriented and, er, business process orientated, er, type software, that…that…that was really what we…we did.

[00:58:35]
Erm, it reminds me of, erm, the development of Larry Ellison, who, of course, was in relational databases and loved that. But erm, then he was forced to…to look at the type of applications that SAP were doing, and he got deeply into the process. So, that's what you did as well—looked deeply into the Workflow processes.

[00:58:56]
Exactly, erm, how you control the arrival and the… you know, sort of—how events, er, can drive work.

[00:59:06]
We're coming up to a seminal, erm, a seminal date—Y2K, we have an enormous spectrum of erm, opinion about Y2K, already in the archives, perhaps you would add, erm, Mandy's approach. Erm, we have, on the one hand, people who say, "What a marvellous way for consultants, for consultants to trouser huge amounts of money." And on the other, people say, "Absolutely essential, look, nothing really critical fell over and that's how good we are." And more than that, it laid the foundation for e-commerce because everybody had really upgraded their software. What's Mandy Chessell's, erm, opinion of this?

[00:59:45]
It's the second actually. I…I think it caused a clean-up of an awful lot of old code, erm, and got people thinking about the future and, er, you know, and there are certain things that we—I mean, we're going…we're going to have another problem with dates, erm, coming up because of the way we do…we do dates, erm, er, in Unix. But, yes, I did think—I think the people who were involved in many, many organisation systems, who dedicated a lot of very detailed work to make, um, Y2K as uneventful as it—you know if we'd done nothing, everything would have shut down. We would have had so many bizarre errors and we… the…the trust in, er, IT would have been deeply, er, shattered [laughs] in the eyes of the public, if we hadn't have done this. And it's one of those things that, you know, when you do a good job, nobody notices.[laughs] 'cause that's the art, I mean, for most systems, when they work, you really don't notice them and it's only when they're failing that you get… you know, you get frustrated with them.

So, I…I think it was important, erm, I'm sure there were people who made money with little value but to be honest, there were an awful lot of people who were very dedicated, er, and went through—and also, didn't drink that night, even though everyone else was—because they were waiting to make sure that everything worked

okay. So, from an industry point of view, I think we should have been proud—we should be proud.

[01:01:17]
Now, Mandy Chessell is moving up this technology path in IBM, not the management path, but the technology path in IBM. And erm, does she see many women around her, or are these women being shed as she goes up?

[01:01:34]
So, erm, in IBM, erm, women tended to be encouraged into the management side, er, because er, communication skills were recognised and you can make, er, an initial first steps much faster. So, it's…there is a lot of, erm, a lot of benefits, but it is then much harder to get to the…to the higher jobs. So, the number—I didn't no…notice it too much. Erm, and initially I was very resistant to any women in technology-type, er, schemes and, you know, I wanted to know—I'm completely fine, I do not need any special support or whatever, I'm…I'm not detecting anything unfair. Erm, but this actually, was my ignorance [laughs]. Erm, and…and most women do not want anything special, they want to be treated equally, erm, and are not aware of the fact that every day, they're act…they're actually not behaving enough like a man to be considered technical. So, it's something even though everything I've done, I still have people asserting that I'm not very technical.

[01:02:53]
[Laughs].

[01:02:54]
And this is—it…it…it [laughs] anybody that I work with, I have no trouble with, but it's people who are on the peripheries, and they are often people who are offering you jobs and things. And…and…and so, yeah, so, f…for many young women, they do not want to know that there is a problem in the industry, and I certainly was…was of that opinion. And IBM wanted to send me to California to go to a Women in Technology Conference, and I said, I didn't want…want to go to that—that sounds…that just sounds awful. Erm, and there was an awful lot of fuss because I refused to go. Erm, but a little bit later—a year or so later, I was invited to one in

Milan, and I thought, oh, this is only 3 days, I don't have to go to America, er, I'm just going to go, keep my mouth shut and—because you know, so, there would be no fuss. And it was extremely important. They talked about the way that, erm, men and women communicate, how politics of where you sit at the table in a large meeting, how you—all the sort of things that were completely, I just had no idea about. Erm, that…it sort of started to open up, that… that yes, I'm different, I do things differently, I don't even know I'm doing things differently. Because often, I was the only woman and so, I couldn't see how different I was, my fellow colleagues knew exactly how different I was [laughs], er, but I could never see it because everyone I looked at was male [laughs].

[01:04:26]
Mm-hmm.

[01:04:26]
Er, and er, so…so, yes, there were very few women, erm, but that was true with my degree, er, and in fact, actually, when I was 15 and I'd made all my 6 selections, I didn't know that girls weren't supposed to do maths and physics [laughs] and I got a horrible shock. I thought there was something wrong with me when I was 15 and suddenly, the number of girls in the class just dropped. And I thought, oh, dear, oh, I didn't know I wasn't supposed to do these subjects. Erm, but—

[01:04:59]
You didn't know you were in the wrong room?

[01:05:01]
Exactly, exactly. So, you know, I didn't get that memo [laughs] and I…and I've ceased to get that memo ever since [laughs].

[01:05:09]
And from your efforts and the other efforts, erm, because IBM has a…a particular programme or several programmes on this—what do you think has moved forward, Mandy?

[01:05:21]
So, erm, I think the…the idea that there are many styles of career that it's not…you don't sort of—it's not like a sort of running up a hill until you get tired and then you stop and that's as far as you go—that people have breaks in their career, there are times when they are really focused on promotion and moving forward, there are other times when they need to pull back for their families. And it's not just children, it's…it's, erm, parents, it's, erm, health, it's different things going on in people's lives. Erm, and I think m…many companies, and actually the…the—again, the pandemic has made people realise that actually, people can function when they're working from home. They—you know, you don't have to bring people into an office and watch them to make them work. Er, and so all of these things have helped everybody because that is the funny thing, the stuff we've done for women in technology benefits everybody because it allows people to be more themselves and doesn't create that single cookie-cutter, you know, this is how you have to behave if you want to be an engineer. Er, so, that…that to me, is…is…is the most important thing, is the opening up, the recognition of different communication styles a…a…and ways of working.

[01:06:46]
Do you not think it needs affirmative action?

[01:06:51]
Still—yes.

[01:06:53]
You think it does need affirmative action?

[01:06:55]
It does…it does, but it's not just gender, it's erm, its…and it's…and this is again what I like, is now we…we've moved—the…the main reason that there was a big focus on women in technology is because it's easy to count. Erm, and it's much, much harder to count different communication styles, as when neurodiversity-type issues. Erm, different cultural and ethnic things. So, when you—the biggest different between people is men and women [laughs] and so, it's a good thing to focus on, erm, in terms

of monitoring, er, what happens in the organisation. It's just very hard to fake figures. So, that's why women in technology is…is important, but all the schemes need to work on allowing people to be as diverse as they need to be, that…that to me is the thing that we need to focus on, cause it's too easy for a manager who is not an engineer, to focus on behaviour, not results. Whereas when you have managers who are engineeringly capable, they focus much more on results, that's what—that's my experiences that I've done very much better with managers who were technically capable and could understand what we were doing. Whereas those who said, well, you're not—you don't seem very technical from—the behaviours I would have expected from a technical-- So, yeah, that's…that's really where I think the future lies in that—focus on—more focus what people achieve and how they work together and that type of thing than, er, on, erm—than on things—

[01:08:33]
So, Mandy, what are the biggest mistakes you have made in your career?

[01:08:38]
The biggest mistake?

[01:08:39]
Yes.

[01:08:43]
Erm, choose—I've got to choose them [laughs]. Erm, often, yeah, I…my nature is to avoid conflict, and sometimes it's unavoidable and so there have been times where I should have fought harder. Because with innovation, you often have to give it time, you have to get it into the right time and place context. And so, sometimes, you keep it hidden back, but there are other times where, actually, you now have to go forward very aggressively with it, and that's the bit I find the hardest. I find people who are being aggressive to me, I let it intimidate me, so, that's…that's my biggest, you know if I look back on where I should have, erm, you know, stood up more and not be—you know, and not…and not allow myself to be…to be pushed to one…pushed to one side—that's…that's probably my biggest mistake. And it's a recurring one and I hate

having to say, "I've done this—I've done this—I've done this" erm, so, yeah, that's…that's…that's the…the pattern of…of…of…of…of, er, failure.

[01:10:01]
Unfortunately, it's not true that a career speaks for itself. Your CV speaks for itself, but then people quite often just won't realise the value of it, despite the enormous contributions you've made to the industry. Er, a couple of last questions—do you mind?

[01:10:17]
Yes, plenty of time.

[01:10:19]
Some people think that AI is going to take over the world and that humans are going to be dominated by it, what does Mandy think?

[01:10:25]
[Laughs] it's always our choice, erm, I mean, what is AI, it's…it's…it's decision-making with data and there's an aspect that we can ultimate, so, erm, choosing when the lights are on in the building, particularly the big building. Erm, that requires understanding of where people are, so, there are lots and lots of, things that are tedious for us to do, but do require, a lot of data to be processed very quickly and that's where AI is good. There are many things that A…AI will never understand the value of humans, erm, you can sort of teach it, but not really because there…there's an irrationality to our, erm, [laughs] to, uh, many of our decisions, um, which is a very subtle combination of different ethical considerations and things and that, I think will be very hard for AI to take over. So, we…as with all these things, we will probably make mistakes, but we need to use it for where it is most effective. Erm, now, the…the real question is where power is and at the moment, we have power in a very small number of large organisations, and that's probably not healthy. Er, because, when you don't know people, you can be more callous in terms of your user technology, erm, and so, we do have to think very carefully about these very large corporations and their power. But at the same time, technology is more available to

everybody than it's ever been, and…and so, as long as there is that open access to technology, then there will always be a counterbalance.

[01:12:14]
So, you cannot embed ethics into systems, is that right?

[01:12:20]
That's…you can embed rules, erm, that, er, have designed from an ethical basis. But everybody's ethics is different, it's changing, we have different opinions. Often we think something is wrong when we're not familiar with it and over time, as we come familiar with it, we accept it more. Erm, so, I would think if you went back 50 years and said everybody is going to be filmed when they're out in the city, you would be horrified, but we're filmed all the time. Erm, and actually, from my perspective, that makes me…gives me more freedom because I feel safer, erm—

[01:12:58]
I don't think Matt Hancock feels safer.

[01:13:01]
Maybe not [laughs].

[01:13:03]
But I know what you mean.

[01:13:05]
Yes, but, I mean, and…and probably for many women, it's been freedom, er, even though it appears to be, erm, a loss of privacy, erm, but actually, it's…it's enabling. So, there—you know, it's very…it's very difficult because there's no such single thing as ethics and we all have our own view, it then makes it very difficult to have, you know systems that are embedded ethics. But you can make them adaptable, you can make them favour certain types of behaviour. You actually could make it dis…because most data that AI is built on is biased because it comes from the real world which is biased. So, you can actually get it to…to actually move away from what it sees in the data and, er, and create a slightly fairer approach. So, you can do

all these things, but it's all about people making those choices to make the system behave. At the end of the day, it's always a person-made a choice to make the system do what it does. Erm, and that's…that's why we have engineers, that's why we need a broad base of engineers, er, because we need a broad base of people making those types of decisions.

[01:14:21]
For 35 years, this very bright, very articulate, and very forward-looking, and flexible person called Mandy Chessell has been inside IBM. Has not the thought occurred to this very bright, etc. person called Mandy Chessell that she should be outside IBM?

[01:14:41]
Yes. [laughs]

[01:14:42]
Why not, why didn't you?

[01:14:43]
Erm, so, erm, well actually, I'm about to retire, erm, and er, the plan is to, erm, to leave IBM, and to spend what…you know, but to…to continue to work on open-source, to work with more effective engineering, maybe to do some consultancy. But yes, this is…this is exactly the right time, 35 years within the co…within the corporate world and now, I am literally on the verge of, erm, moving to more of a…you know, sort of, er, being an individual, and er, working outside of IBM. So, it's going to be a big change for me.

[01:15:23]
It's going to be a shock [laughs]

[01:15:26]
[laughs], er, but yes…yes.

[01:15:30]
Well, we wish you the very best.

[01:15:31]
That's….that's…that's about to happen.

[01:15:32]
And I'm sure that the, erm, the archives will want to, erm, come back to you in 10 years or so, Mandy Chessell, and find out how you've done, erm, in your complete and new role of being an individual outside IBM. Your career is exemplary, fantastic. And now, I have a little time to read some of your professional recognition. So, you have already, erm, a CBE, you are Honorary Doctor of Science at the University of Bath, Honorary Doctor of Technology, University of South Wales, Honorary Doctor of Technology, University of Brighton, erm, a Fellow of the Women's Engineering Society, Honorary Doctor of Science, Plymouth University, etc., etc. Mandy Chessell, thank you so much for your contribution to the archives, it has indeed been an inspiration, thank you.

[01:16:24]
Okay.

Interview ends: 01:16:26