#  School C Teacher 4 – 31 Jan - 2017

Speaker 1: Got interested in the computerization of the print process, but it was mainly about things like on-screen layout and all that sort of stuff.

Speaker 2: [inaudible 00:00:09].

Speaker 1: Mm-hmm (affirmative).

Speaker 3: Where was that?

Speaker 1: Sorry?

Speaker 3: Where was that, was that up here?

Speaker 1: No, I was working in the east end of London.

Speaker 2: Fleet Street-ish that whole ... So, you were down there when that whole ...

Speaker 1: Yeah.

Speaker 2: All the strikes went on, because of the movement to digital publication, mainly.

Speaker 1: That was really what ... I went through the Wapping Dispute, and then what was clear post Wapping was that there wasn't really a glorious future in printing.

Speaker 3: What you're saying ...

Speaker 1: It's like, I had a choice really. I considered doing a graphic design course, because that was what I was getting into with the odd screen layout, that sort of stuff. Or, teaching, retraining as a teacher. What I'd been doing in printing in the meantime, is I'd been working with trainees on YTS schemes, coming into printing. I quite enjoyed that, so I thought, "I'll retrain as a teacher." There was no IT in those days, there was ...

Speaker 2: There wasn't a national curriculum either.

Speaker 1: No, so I retrained as a design and technology teacher. Hello Matthew. Log on and I'll get you started in two ticks. I trained as a design and technology teacher, the national curriculum came in while I was retraining as D&T teacher. IT was the fifth strand of design and technology. It was being ...

Speaker 2: What were the other four?

Speaker 1: I can't remember now. I came into schools just as a D&T teacher, where there was already one or two pioneering O-Level computer studies courses going on. There was no IT departments. I was quite lucky because ...

Speaker 2: Is that the late '80s or the mid '80s?

Speaker 1: No, that's the late '80s.

Speaker 2: Late '80s.

Speaker 1: I was quite lucky really, because with the national curriculum, very quickly the national curriculum hit problems. In the Dearing review, IT became a separate subject in its own right. Schools suddenly had the need to staff an IT department and the need to develop an infrastructure. I was quite fortunate, I was working in the biggest school in London, my head of department was a techie.

Speaker 4: Is this computer science?

Speaker 1: It is indeed, would you like to join us?

Speaker 4: Yes please, has it started yet? I mean today.

Speaker 1: No, we're just about to start.

Speaker 4: Okay.

Speaker 1: Just log yourself on, I might have to give you a little bit of an introduction.

Speaker 4: Okay.

Speaker 1: Yeah, so for my first sort of eight years of teaching, really, we were teaching ourselves networking. We built our own networks and all that sort of stuff. I was quite fortunate to be in a school where ...

Speaker 2: Were you doing your work ... I was speaking to Wendy, up in [inaudible 00:03:53], and she was saying when she was teaching networking, she ended up also setting up a lots of the network up in the school, because she had the knowledge to do it. Was that something that you found as well, where you ended up helping? Not only was this a new subject, it was also a new change within the wider school that was happening at the same time?

Speaker 1: Well, there was no one else, there were no technical people, so you did it. We literally, we laid our own cables and everything. You know, yeah.

Speaker 2: You were the technical people as well as teaching IT?

Speaker 1: Oh yeah. Yeah, that continued right the way through in most schools, right through into the 2000's. I'll just get this [inaudible 00:04:41].

Speaker 2: Yeah, yeah, go ahead.

Speaker 1: Folks, I wasn't here last week.

…

Speaker 1: Right, so I'm going to ask you a bunch of questions. Some of them will be relevant. Do you have any questions?

Speaker 2: No.

Speaker 1: You probably heard me introduce my research to enough people at this point, that it all seems fairly old hat. So, you were talking about, you're doing networking, and then fast forward. When did you start here, or what was that?

Speaker 2: So, I moved up north in 1998. I moved from working in the biggest school in London.

Speaker 1: Wow.

Speaker 2: Where there was more money and, more resources available, which meant that we built a house. By the time I left London, we'd built the whole school network, and I taught myself the basics of systems, admin, and so on. When I came up here, you know, that infrastructure I take for granted wasn't replicated up here at all. I went to Settle High School. That's where I first worked, just for a year. I spent that year doing essentially the same thing.

Speaker 1: Setting up building systems.

Speaker 2: Yeah, and then the vacancy came up here. What was immediately apparent at Settle is that small, rural schools couldn't offer the subject diversity that I'd taken for granted. One thing that I'd been doing, as well as setting up the infrastructure down there and introducing IT as a national curriculum subject, is we also taught A level computing. So, I'd been teaching A level computing in London, which again, I fell into by accident. I fell into that because my head of department went off on a long term illness, and someone just had to pick up the A level group.

Speaker 1: That was in the mid '90s?

Speaker 2: I started teaching that, yeah, so, 1994 or '93 when I first started teaching that.

Speaker 1: When was, you mentioned the Dearing Review. When was that?

Speaker 2: That would have been about that time. You'd have to look it up. The national curriculum came in, in '91. It was pretty clear it was unworkable by very early on, and there was a boycott of sorts that effectively led to the Dearing Review and a rejigging of the National Curriculum significance.

Speaker 1: That makes sense. In literature, people are talking about the National Curriculum in terms of '89, so it must have been ...

Speaker 2: Yeah, it was already, it was being discussed in '89, and I think it was implemented in the start of '91.

Speaker 1: Late '80s, got it. Then the Dearing Review came not long after that. That's when ICT became a subject.

Speaker 2: Well, it was IT. So, it was IT, and it changed its name to ICT sometime later, around about '97, '98. Again, no I'll say this, I might be getting confused now between when the Dearing Review was and when the name change is, but the name change certainly came later because, you see, the early IT programme of study had a much closer relationship to what we'd understand as computing than the subsequent ICT that came in. In a sense, what happened in the '90s was loads of people, shall we say, got on the IT bandwagon. Many of them embraced it because they were themselves beginning to get computers at home and things like that. Their experience with computers was just as software users. They felt what we needed to do was just, introduce the software tools in education.

 There was a big move that IT could be taught across the curriculum. So, much of my time in the '90s when we were introducing IT, was defending the notion of it being a separate subject. This move to ICT, sort of reflected this idea that really it was much more about getting this empowering technology into education across the curriculum everywhere, than it was about an academic discipline at a foundational level.

Speaker 1: Do you feel there was a skills' loss because of that?

Speaker 2: Well, no, but there wasn't much of a skills base in the first place. What there was, was that you lost the definition of a subject to the extent that it had never been properly defined. It had never really been properly defined. There was always a tension between that idea of software use and an underlying academic discipline. The underlying academic discipline had been much more apparent in the early days of O level computer studies than when IT came in because once IT came in, every school had to teach it. Anybody, it was usually the business studies teachers or somebody who had a computer at home. The cop out was if you didn't have any teachers to do it, you taught it across the curriculum.

Speaker 1: Yeah.

Speaker 2: I remember, you know, in English, you'd get them in the computer suite, and you'd do a bit of word processing about the holiday. In business studies, they'd do a little spreadsheet. In math, they'd do a bit of Logo, you know.

Speaker 1: So, that was when it moved to ICT, in the early stages of the ICT?

Speaker 2: Well, that was in the early stage. Then that really gained momentum with this notion of ICT, Information Communication Technology. The big thrust of everything then was about, because this was the start of the internet coming in as well, so it was all about using this newly emerging infrastructure as to assist in the development of education as a whole.

 Actually, the period just after 2000 on, was really the time when the big money came into education with the Stevenson Report and the National Grid for Learning. The NGFL, National Grid for Learning, funded, there was huge funding went into schools, to get schools to build the infrastructures, the networks, that have internet connectivity. You then had access to the internet for children.

Speaker 1: In most schools?

Speaker 2: Yeah, well, that then took place through that period, probably from about 1999, 2000, through until ... First, again, you'll have to look at the dates, but I think the Stevenson Report was probably in '99. Significant funding came in with the new curriculum in '97. How it got allocated took several years to work through, but the basic developments there were the broadband consortia, that provided the internet access and money to schools to provide the networks that connected into that. That was when you began to get the developments of technical support and that sort of stuff. Up until then, it had just been a handful of enthusiasts.

Speaker 1: Underlying that process, it sounds like ... I was aware of, but it's interesting to hear it as it's lived. It seems like there was also a real change, a movement of emphasis. So that early on, there was that early IT in the late '80s, was information technology, new computers, not a huge number in homes. Probably some people are going to use these, so they need to know how they work. Whereas, it feels like with ICT curriculum in '97, it was far more, everybody is going to need to use these things in some way. There's kind of a shifting of what's the purpose of all of this.

Speaker 2: Yeah, but the shift was from the idea of computer science, if you like, or a discipline that underpinned the technology that was emerging to one of using this to, and the buzz words then were transform education. Essentially, you were suddenly connecting children to the internet. Suddenly, they've got these huge, vast reservoirs of educational opportunities to tap into, in all sorts of subjects.

Speaker 1: How typical do you think, I mean, it's interesting that it's already shown up a few times in this research, but how typical do you think someone like yourself was at that point who was kind of self-educated in terms of computing, and come into teaching to do DT or something else and then kind of had self-educated yourself in terms of the computing, or were there people coming in, kind of, with computing backgrounds coming in to education?

Speaker 2: Well, there might have been the odd one or two odd people. Invariably anyone who had a bit of a computing background tended to be in a science or a math department. There weren't jobs.

Speaker 1: Right.

Speaker 2: Basically, anyone who was in teaching with that sort of background was in usually a math or a science, or possibly D and T. That's where, these early people emerged, out of these disciplines. When there suddenly was the need to teach a subject, these were the people who stepped forward and became the first heads of the IT departments.

Speaker 1: People very much learning themselves. It wasn't people who had done computer science in university, for example.

Speaker 2: There might have been the odd one, but you see, there weren't many computer science courses in university.

Speaker 1: There weren't any computer science courses.

Speaker 2: Again, I mean, so all of this stuff is emerging over time.

Speaker 1: That's one of the things I've learned through this research process, is that actually if you start looking at computing curriculum or introduction of it, the first stuff you find is about that introduction to universities. You know, the first computer science PhD's are actually not that long ago. I think they were mid-60's, something like that. This is not a long history. It's taken a long time for computer science as a discipline at university level to be pinned down to something that everyone agrees on.

Speaker 2: I mean, you saw when I was at university in 1979 doing this FORTRAN, that was punch cards. We took our punch cards, you never got anywhere near a computer. There was only the one mainframe there at Warwick. You delivered your cards to the thing, it ran overnight, and you went and collected your print out in the morning. So, I mean, yeah and although there were degrees in computer science, that was the limitations of what there was. I suspect, I've no idea about numbers, but I suspect the take up of these things, it was probably a very early, emerging discipline as well.

Female speaker: We were in the next room so when your finished [inaudible 00:11:27].

Speaker 1: I think we met last time, didn't we, when I was here?

Female speaker: Yeah.

Speaker 1: You interview so many people for a project like this.

Speaker 2: I can probably dig out some good ... I mean, I've written stuff in the past about all this early history and that sort of stuff.

Speaker 1: Tell me about, again, from your perspective, I feel like we've kind of gone up to about 2000, mid-2000s.

Speaker 2: The big turn here is the turn of the century, huge money going into schools for the National Grid for Learning. That giving further emphasis to a cross-curricular use of computers as a transformational thing. Allied to that is probably the other most significant development, in terms of the development of the subject, which is that you have vocational qualifications, principally the GNVQ that were introduced, which were-

Speaker 1: That was under labour, wasn't it?

Speaker 2: Yeah. The GNVQ's came in. There was the advanced GNVQ, whatever it was called. You could get either. You could do a vocational qualification in IT that was worth either one, two, or four GCSEs. This was also, the late '90s had seen the introduction of league tables. All of a sudden, schools realised that league tables were based on five GCSE passes or over, and you could get four of them by passing the GNVQ IT, and it was a Mickey Mouse qualification. Thomas Telford, the head teacher at ... Is it Thomas Telford School down in, well it's in Telford. I'd have to look it up. They topped the league tables several years running because they basically pushed everybody through the GNVQ IT. This quickly became the recognised way for struggling schools to boost their standing in the league tables.

Speaker 1: So, we're talking late '90s?

Speaker 2: Well, this really gained significantly, if you look in the Royal Society Report when that came out, 2001 to 2005, that became a stampede. The numbers shifting from any IT qualifications that perhaps had a more academic content, something that was entirely assessed on your ability to do things, by outcome and there were online results as marked by Thomas Telford that you just bought into. You didn't even need a teacher to teach it. They just did it, churned it out, and they got four GCSE's.

Speaker 1: What were those skills that they were learning?

Speaker 2: There were very little. What you got, essentially you got the devaluing of the currency to the point where it had rapidly become, by about 2005, it was obvious to anybody who'd been around at the time was that as a subject, it had become fundamentally devalued. What was the point of employing these sort of people.

Speaker 1: When, you could just buy the resources.

Speaker 2: When you could just buy off the resource. They could do it with a supply teacher almost and get four GCSE's.

Speaker 1: Do you feel like, it sounds like one of the things that happened with those league tables, the driver for kind of teaching specifically IT, but really across the subjects, was about where you sat on the league tables, not about the skill set the students left with or what they were prepared to do.

Speaker 2: I think there were two things here. From a school's point of view, league tables were increasingly becoming the big issue. That became the driver for a huge shift, and a massive take up in GNVQ IT. So actually, that meant that you did actually get quite large IT departments, but basically the people didn't have to have any knowledge to teach this stuff. But at the same time, there was huge money coming into education powered by an increasingly big gravy train that had endless consultants and LEA people.

 So really, what you got was you got a lot of the old, the sort of people I started teaching with. Many of them went off into positions of local authorities or consultancy work, project management for the National Grid for Learning, broadband infrastructure, all this sort of stuff. They left teaching, so you actually had, a lot of those initial pioneers left and got more lucrative careers elsewhere.

Speaker 1: Left teaching.

Speaker 2: What you were left with was a fundamentally, a bunch of people who generally had come into IT to fill up their timetables almost, ending up becoming-

Speaker 1: What you ended up with was a real de-skilling of the workforce.

Speaker 2: Yeah, yeah, oh absolutely.

Speaker 1: The people passionate about the workforce.

Speaker 2: Yes. Yeah, although at the same time, you also, because this was a subject that was increasing, you did also begin to see PGCE people coming through. So, you did actually have the first people who were qualified as IT teachers, or ICT teachers coming through having completed PGCE's. Again, I would question what was covered on a lot of those PGCE's in those early days. It was-

Speaker 1: Again, if it was fairly Mickey Mouse at the qualification level, what was the PGCE, and was it being taken by people as an easy subject. I could be a math teacher, but that's quite hard. I'll do ICT because that's just typing.

Speaker 2: Yeah, and there was a fundamental move, overwhelmingly the most important thing in terms of outcome was office skills as being the touchstone for this. At the same time, there was all this futuristic talk about education for the future and that sort of stuff that drove access to the internet.

Speaker 1: So, there was a rhetoric around education of the future. In the reality, what was being delivered was office skills, basic office skills.

Speaker 2: Yeah, and then in other subjects, because again, there was a huge tranche of funding went into what was called the new opportunity, NOF funding, which went into re-skilling teachers in other subjects to use IT. $450 million, it was a ludicrous amount of money went into some Mickey Mouse courses for them teaching them how to surf the web.

Speaker 1: $450 million, what time? What sort of time?

Speaker 2: This would again have been probably 2001, that sort of time.

Speaker 1: Right. So, that's for training though.

Speaker 2: That was for training, but then there was also curriculum credits for buying software, which meant schools, of course, had ring fence funding, which meant you had to spend it on badged educational software.

Speaker 1: Because we still see some of that software in use, don't we? Software that was bought with that money.

Speaker 2: Yeah, again, you've got another legacy here of, sort of wasted money where because there was the funding there, you got every Tom, Dick, and Harry producing crap educational software.

Speaker 1: And hardware.

Speaker 2: Yeah, less so hardware. The money was there for hardware, but really hardware was just standardising your own IBM clones and standard PC. So there was less wasted money there than, it was the curriculum software. So, I'll give you a good example. Whereas, when I started teaching, in math, they did lots of exploratory work in Logo. Now, you'll find loads of schools use something like My Math software, which is skill and drill. So, you've got this fantastic explosion of software that, Seymour Papert, when he wrote Mindstorms, which was so prescient, talked about even in the early days, he talked about the difference between his idea of what the power of the computer was in terms of developing children's capacity to think, and what he said was the emerging educational software market.

 He talked about, he asked the question, is it that we want the computer to programme the child or do we want the child to programme the computer? Increasingly, what you were finding was that where cross-curricular use of IT was being embraced, all that was happening was classes were being marched into a computer suite. They were being sat in front of these skill and drill software packages, and were just clicking through. So, essentially, most educational software mimicked 19th century chalk and talk teaching methods, except it was just on a screen.

Speaker 1: So it was on a screen. So, what we see, then, is an elimination of the role of the teacher, but not into a role of facilitator, but into a replacement by a software package.

Speaker 2: Yeah. That, again, is one of these interesting sort of threads that is steering development here. From the policy level, there was always huge interest in this because teaching is a labour intensive and expensive job, and there's always teacher shortages. So, from a policy level, the transformational impact of IT in education was as a replacement for the teacher, and for-

Speaker 1: We could be kind, and say as a supplement to the teacher, to increase efficiency, teacher efficiency.

Speaker 2: Yeah, but increasingly, it was skill and drill type software that was written around the ability to assess, so you could generate marks, so you could then get all sorts of data crunching that could go on to demonstrate progress. When in reality, what you were doing is you were assessing what was easy to assess, rather than looking to develop something that assessed something that aided education.

Speaker 1: When you can't measure what you want ...

Speaker 2: You measure something else. That still remains a key.

Speaker 1: What's interesting about that story you just said is that is now almost taken for granted. Things like being able to do quizzes at the end of each lesson. It's increasing, and it's getting better, but doing quick quizzes at the end of each lesson is only possible if they're marked by a computer, by a software package.

Speaker 2: Yeah.

Speaker 1: So, that level of using computing to replace or supplement teaching has happened subtly and completely, where it's just taken for granted.

Speaker 2: I mean, I think you see a lot more of that in primary schools perhaps than secondary schools, but yes, all of this is a legacy of pumping that sort of software into -

Speaker 1: What's different in primary schools. I think one of the differences in secondary schools is you take for granted, and again, this is a really interesting part of the discussion, a degree of computer competence. You take for granted that you can have a class of 30 come in and sit down at 30 computers, and they'll all be able to work with them.

 Realistically, except for five and six in primary, that isn't necessarily as easy for a load of reasons. I think primary schools, the teachers, especially smaller ones where you don't have individual computer groups, they don't rely on computers for assessment because too many of the kids just can't access them. They know the kids couldn't be tested like that, where once you get to secondary schools, you know you can test on the computers because they can hopefully all use them.

 That's another part of this discussion, is that alongside these education developments, there's what's happening at home. That seems to be doing some sort of weird curviness where there was very little skill at home, except in a few households I would guess, to most households having a computer with a screen, but probably only one, to kids now having tablets where they poke them. Being pokeable.

Speaker 2: Tablets, yeah. Again, if you're identifying different things, so I mean again, one of the early impetuses for IT office skills was actually most people wouldn't get that access anywhere else. You had to do it in schools. That had a certain validity to it. Part of it was that it was a good levelling up for less formally educated.

 But again, your growth of home PC use and then ubiquitous tablet, that's really only a 12 or 13 year history. Again, you can probably get stats from somewhere, but certainly up to 2003, 2004, lots of homes, the digital divide was a huge talking point in terms of strategic IT addressing. You couldn't set homework on the computers because nobody, half your class wouldn't have access.

Speaker 1: You still see, interestingly, the expectation, and I find this because of my perspective, really strange as a parent, you still see a requirement to bring things in that are hand-written. Because, it's still almost that acceptance that not everyone will have access to a word processor at home. Or it's even where you give kids laborious writing out a sentence by hand.

 So, we're up to the mid-2000s. When did CAS come together, and what's that history?

Speaker 2: CAS came together ...

Speaker 1: In my mind, we have a bunch of self-taught enthusiasts who had networked their schools, knew a load about systems, but also computing in general who were teaching computing at a few different levels until the mid-90's. IT then became ICT, and started focusing on usage. Some of those people -

Speaker 2: And with the exception of A level computing, there were no computing qualifications in school-

Speaker 1: at any level-

Speaker 2: with the exception of A level, so there's nothing at GCSE other than IT.

Speaker 1: Right. You have IT at GCSE, and you had the GNVQ, which was A level equivalent?

Speaker 2: No, that was GCSE.

Speaker 1: GCSE.

Speaker 2: Well, and then you could do an advanced GNVQ.

Speaker 1: Those things -

Speaker 2: And you also had an increasing merging of A level content between the IT A level, and the computing A level, so actually in terms of practical project work, programming effectively dropped out of A level computing and instead people were invariably for project work, were using visual basic applications and building databases.

Speaker 1: That's interesting to point out, which I had missed is that programming was there earlier on, using I'm guessing BASIC?

Speaker 2: Yeah, BASIC and Logo were probably the two. Well, BASIC, Logo and Pascal.

Speaker 1: Yeah, there was a smaller gamut of programming languages at that point. But then with the IT, except maybe with their A level computing, you weren't doing any programming I'm guessing.

Speaker 2: Yeah, there was virtually no programming unless you kept it going. So, I mean, you could keep programming as an identified element in the national curriculum had gone. Increasingly, you had this shift to very much a focus on office applications with a view to the final outcome being, the vocational qualifications that focused on the office skill.

Speaker 1: The other thing I can see within that history is that people like yourself had a wide range of languages, not necessarily because you were experts at languages, but because computer languages were developing as a process, so you've learned FORTRAN, which is a fairly basic language. Through that learning several languages, you're learning a range of computer science skills, whereas I think now we have a number of fairly stable languages that do a wide range of things. There's not as much of a need for somebody who's teaching or has been teaching for the last ten years. If they know something like Java or HTML or Python, they can still teach those same ones without having to update themselves, necessarily in the way that somebody like yourself has had to do.

Speaker 2: I think there's more to it than that in terms of people teaching programming. I think one of the worrying developments of the new curriculum-

Speaker 1: Yeah, that's what we're getting to.

Speaker 2: CAS emerged if you like, in response to this growing crisis in the direction of IT.

Speaker 1: When?

Speaker 2: We had our first meetings at the back end of 2007, or was it 2008? It was early 2008, back end of 2007.

Speaker 1: How did that form, how did that group?

Speaker 2: It came together because there were, like you say, there were individuals who were ploughing their own furrow and were still flying the flag for things. Scratch emerged in about 2004, 2005. Here, I'd been introducing Scratch, I'd been introducing Game Maker, [inaudible 00:30:31] University, which was wonderful. In a sense, whilst things were clearly going wrong with IT, there wasn't a coherent view as to what was wrong. What there was, was lots of people who held on to the older ideas of what IT should have been, who were experimenting with newly emerging stuff like Scratch. That was the first real block-based language beyond LEGO MINDSTORMS and Game Maker, things like that.

 I mean, I remember doing it, to give you a sense of the balance of forces then, I did a meeting. Cumbria used to have its ICT conference every year, big thing at Rheged. You used to get old school [inaudible 00:31:24]. It was a big thing. Catherine the Great, Lou Gordon, Steve Mosser was the adviser, would give a big speech about transforming education.

 I did a meeting about getting pupils to think at key stage three, and about the resources that were available. I thought Scratch and Logo and that. I had I think six people at my meeting and there were hundreds for this keynote speaker who was going on about his, some web resource, said it was improving literacy and whatever. There was hardly an ending.

 Now, but what we had had from 2004, I think was that, or 5, we'd had a change in A level specification, and the A level specification had put clear blue water between computer sciences as a discipline, or computing and the ICT A levels. So, programming was coming back in at A level from that sort of time onwards. When there was a change in specification, that I set up an online forum for people to support teachers in teaching the new A level spec. That just grew by word of mouth to 350 people.

Speaker 1: Okay.

Speaker 2: So, there was just me off me own to set this thing up, really, initially as a support strip for people in Cumbria that grew by word of mouth, and the emerging [crosstalk 00:32:50]

Speaker 1: It grew beyond Cumbria, yes?

Speaker 2: We ended up with 350 active A level teachers around the country. It was clear that lots of them were dissatisfied with things. There was the initiative company. Science for Fun, was launched by Queen Mary University of London, which meant that a magazine was coming into schools. I went down to a ... I got contacted by Roger Boyle who was professor computing at Leads University, where there was growing concern in higher education about the dramatic drop intake of computer science, because following the post war boom, intake have plummeted.

Speaker 1: We're also talking about ... If we're talking, the other part of this history then, is late 2000s we had the digital bubble,

Speaker 2: Yeah.

Speaker 1: Which, had bubbled.

Speaker 2: Yeah.

Speaker 1: So that in the early '90s it looked like it was a great ... and even in the early 2000s you had Amazons, and you had Yahoos, and you had Netscapes, and they were all kind of bubbling up, and then that burst. So suddenly it looked like not such a great idea to study computing.

Speaker 2: Yeah, but you also had in higher education the mirroring of what was going on in school to some degree. There were loads of ICT degrees being offered, which were much more skills based. Animation, visual effects, all that sort of stuff you have. So, but certainly there was a growing concern in higher education by about sort of, you know, 2004 or 5 that that bubble had burst.

 Recruitment was drying up and that was mirrored by concern in the commercial world that they couldn't get people because they were coming through, the pipeline had dried up. I went down to a recruitment fare down in Cambridge on the invitation of Roger Boyle, that was aimed at the final year undergrads who were predicted to get firsts in computer science. All the big players were there. Intel, Microsoft, you know, Google and so on. They were all fishing for potential PHD candidates. And offering finances to finance them because they just couldn't get researchers.

Speaker 1: Yeah.

Speaker 2: So, you really just happen to get just at the same time you got you know, a body of people in teaching saying something's going wrong here, a body of people in higher education saying something's going wrong here, and a body of people in the commercial world saying we've got a problem. It was literally at that recruitment fair that we had a meeting where, I said well I've got this network of 350 people and Pete Adickman, who at the time was doing outreach work at Glasgow University, who now works for Google, he said "well we get into a lot in the Glasgow schools and this is the stuff we've got. We find it's well received. Microsoft Research were there." CAS was born out of that and had its first meeting, which was a dozen people at that sort of time.

Speaker 1: That sort of time being 2004 or 5? [crosstalk 00:36:04]

Speaker 2: No so we've got a period 2000 and sort of 4 to 7 where you've got a few things getting off the ground. Computer science for fun Green Bay University, computer science inside the Glasgow University outreach work. You got my A Levelling forum in schools. All of these sort of things just getting a group of people around them.

 Then CAS really was the brain child of Simon Peyton Jones at Microsoft Research. Computing at school, in a sense, started off really with Simon Peyton Jones and Simon Humphreys who taught at Hills Road College, sharing the same concerns about a disconnect between what he was teaching, they were the big A level college that fed the Cambridge universities. What they were teaching at computing A level, and the fact that there was nothing coming through to A level with any programming experience or computing experience. So, out of that, there was literally an ad hoc gathering of people from, which the CAS group was born. It started off, literally, a half of a dozen people.

Speaker 1: Yeah. But doesn't people ... what I'm trying to get a sense of, and what I'm beginning to get a sense of, is where those voices were coming from.

Speaker 2: From those three constituencies. You literally did have input from higher ed and from the big commercials. Principally Google and Microsoft were the two big; Microsoft Research and Google.

Speaker 1: Yeah. But what's interesting about those two, especially Microsoft Research, it's not really Microsoft, in a way.

Speaker 2: No.

Speaker 1: Microsoft Research, their emphasis is computer science education, computer science research. And yes, there is a commercial undertone to it. But really what they're doing is very similar, or far more similar, to what's happening in the academy, in academics. So that they're somewhere in between, and Google as well I think particularly at that point, but even still so, are as interested in cutting edge research as they are in a skill. They want the people who are going to come up with the new ideas, not because they're going to make lots of money necessarily, but because they're the new ideas.

Speaker 2: Yeah so it was, and I suspect the other big player on the commercial front then, was Ian Livingston, the next gen report. So it was the gaming world.

Speaker 1: That's really interesting. Next gen comes in lots, and I see, and I haven't read it in detail because its quite long and dry. But, they seem to have been forgotten, in some ways, within the way computing as a discipline has shaped up into the curriculum. The needs in the gaming world seem to have been less emphasised than necessarily ... and when I say the needs, I mean the broad needs. So there's a training of programmers potentially but-

Speaker 2: Well you see, I don't know. I mean I think the needs of most of the people that have a vision for curriculum change aren't currently satisfied by the computer curriculum.

Speaker 1: Tell me about that next step. So you've got CAS and-

Speaker 2: So, essentially what you then have is a period between 2008 say, and when Gove made his announcement that there will be a new national topic [crosstalk 00:39:46] which was 2012 was it?

Speaker 1: When did BCS come in and fund?

Speaker 2: Oh well BCS came into CAS very early.

Speaker 1: Yeah. But, what I missed, because I was single, because it's not obvious, is that CAS basically formed before that, before BCS came.

Speaker 2: Only just. I remember the first meeting.

Speaker 1: The people involved kind of formed organically, and then BCS got involved.

Speaker 2: Well, it wasn't that BCS got involved, it was that BCS were invited in. What we were debating was, you know, if you want to try to formalise this as group, what's its aims and its objectives and etc. You've got to remember that at this time the subject association for IT was a body called Mace. Mace had been, although they called themselves a subject association but really the home for a massive layer of what I call the flotsam and jetsam of the teaching profession. These consultants and self employed people who float around and do anything but teach. There are so many of those in the IT world. They essentially, they become embodied in this organisation called Mace. So the big debate in CAS was, do we become part of Mace, which spoke for the professional, claimed it did, or do we see them as an impediment to progress. The argument wasn't no, look actually, we have far more clout. You see, Mace had a history of swallowing up smaller, little groups, so we very much wanted to keep that, initially, at arms length whilst being cordial with them.

 We felt that actually the most important thing was to have representation of the industry, the BCS and of higher education so that the Council of Professors and Heads of Computing. So, we got that sort of representation and BCS basically put some significant funding our way, along with Microsoft Research, which was then matched with Google and a few others.

Speaker 1: But in a very simplistic way, which makes them ... this is a presumption I've had wrong since they started and makes so much more sense now. In a very simplistic way, CAS used BCS far more than BCS tried to use CAS.

Speaker 2: Oh yeah, absolutely. Yeah, yeah.

Speaker 1: You guys used CA ... BCS as a way of getting some funding and getting a bit more clout. But, it wasn't, and this is how I imagined it initially when I'd realised that CAS was funded. It wasn't that BCS said we need ... industry needs more people who have these skills. How do we do it? It wasn't that sort of process. It wasn't that sort of top down process. It was a bottom up.

 We've got this organic meeting of people all interested in this stuff. We need to make sure we have some clout, so we have some influence. Which, strings do we pull? That's a good one to pull and that will give us[crosstalk 00:42:53]

Speaker 2: And essentially, because of the fortuitous nature of the people that we're around in CAS, at the time, you had people who ... what was clear was that on a grass roots level you were tapping a scene of discontent, and there were lots of teachers who were open to these ideas. So we were well represented on the shop floor if you like, or at least we had growing representation once we got on a good web presence and so on. And then what was really important was that, because we had links with the Cambridge Universities and Microsoft Research, and those sort of people, was that they really knew how to pull the policy levers. So, I'd say that really quite influential in all of this was, were people at Cambridge University and Simon Peyton Jones in recognising that if you wanted to get the ear of government, you needed the BCS.

Speaker 1: Yeah.

Speaker 2: It was very much, it was that way round.

Speaker 1: Then BCS sort of approached government with what does industry need but actually that industry need wasn't necessarily coming from industry.

Speaker 2: Yeah. The early pioneer of industry need was Ian Livingston and next gen. They were very vociferous, again, you got different people sort of articulating their own lobbying positions and so on.

Speaker 1: So, we have CAS and we have BCS and then we have Gove. So the next two things in my mind, the next two key events are Gove announcing changes in the curriculum but before that, I think there's the Eric Schmidt speech. I'm not quite sure on which order, but I think there was the Google speech that said "there isn't enough computer science in schools." Which, again is a really interesting talk that's misremembered if you look at the reporting, because he also said the greatness of, the strength of the English education is that its cross curriculum, and you need people with kind of lots of skills. But, there isn't enough people to fill jobs at Google.

Speaker 2: Yeah, well I mean, that was the MacTaggart lecture of 2011 if I remember right. The summer of 2011. Gove made his announcement in January 2012 and Gove, really made his announcement on the back of knowing that probably the most significant things that Royal Society report, which had been in train for the previous 18 months to two years. That was published in January 2012. So in a sense Gove preempted the findings of the Royal Society Report by making his announcement, but knowing more or less what he was going to recommend whilst he was here. So, it was his way of stealing the headlines. So the Royal Society Report really was should the title shut down or restart reflected the fact that from about, through this period from 2007 onwards, Ofsted had woken to the fact that schools were milking the system, and abusing vocational qualifications to play the lead [crosstalk 00:46:09]

Speaker 1: And what interesting-

Speaker 2: and therefore there was significant pressure growing there was a really realistic prospect that you would shut down IT as any sort of entitlement. Which, is why it was called Shut Down or Restart. There was a recognition that it was in crisis. Actually, this was an answer to say no, it needs to actually be restarted. We need to restart it by rediscovering the principles on which it's founded.

Speaker 1: And what's also interesting in some ways, and I recently did something that I should have done a long time ago, is I went back, and I looked at the ICT curriculum from 2007, and comparing it to the 2014 computing curriculum. It's incredibly detailed in some ways. I mean just based on length, one is ten pages and one is three pages. There's a lot more in there and there's a wider range of kind of, doing things creatively, having a wide range of, if you want skills, versus the computing curriculum, which feels like it's a lot of concepts but not a lot of guidance of how to get to those concepts.

Speaker 2: Yeah, there's no guidance at all.

Speaker 1: And none of that, again talking history, none of that 450 million for training, just here's a bunch of concepts that you may or may not have heard of, and we want the kids to know these concepts by these points.

Speaker 2: Yeah. But, I think it wasn't just the IT national curriculum was more detailed, but it was supplemented with what was called the ICT Programme of Study. Which, I don't know if we still have it, but it was file, after file, after file of it.

Female speaker: I'm doing bus duty, so I won't be back in until about 25 to, yeah?

Speaker 1: That's fine. Yeah. I've made them talk again.

Female speaker: Every single one, you said, only about a half an hour, I think every single one of us. None has been less than an hour, have we. So, you said "only half an hour and then, an hour later.

Speaker 2: But you see you have the QCA scheme of work to supplement the national curriculum, which came out in about I'm guessing 2001, 2002.

Speaker 1: Okay.

Speaker 2: No, that was ... I mean if you think the national curriculum was detailed, the QCA scheme of work then gave you a three year scheme of work, literally lesson by lesson, and a three part lesson for every lesson. That actually had a terrible impact on IT teaching because it reduced, again it reduced-

Speaker 1: It deskilled the workforce.

Speaker 2: Yeah, it deskilled the workforce. Nobody thought about what they were teaching or why, they just delivered the QCA scheme of work. I think when used to have 14 folders of stuff up here. I mean, I just never [crosstalk 00:49:00]

Speaker 1: And, if you were talking about 2001, 2002 I bet you would have had a lot of like overhead projector thing, none of these kind of PowerPoint's.

Speaker 2: Well, I mean projectors were beginning to come in. But it was ...

Speaker 1: Transparencies that you could write on.

Speaker 2: But again the point is that you had ... When I came into teaching, the whole emphasis was on teachers developing their own stuff and thinking about what they, before the national curriculum you see, you develop your own syllabus as well. So, you actually thought about what you wanted to teach and why you were teaching it. The whole deskilling process got to, really reached its zenith with the QCA schemes of work where all you have to do as a teacher, in any subject, was pick up the QCA scheme of work and deliver this one on a Thursday morning.

Speaker 1: Because the problem they're trying to solve with a hammer was, there aren't enough teachers and its hard work, so how do we ... and in some ways the other bit of that is that its time, it takes a lot of time. So, how do we reduce the time resource burden on teaching, so we can teach more people because we have a teaching crisis?

 The other, kind of back with the ICT thing, you can see the frustration in some ways, almost probably of, well we were spoon feeding it to them and then we had the shut-down-and-restart thing. If we give them that much guidance, and that much training, and that much support and it still fails, where do we go from there? You can almost hear that argument, of someone tearing their hair out and thinking well what can we do now because we've done everything we can do. We told them what to teach, what more do you want?

 So then we get to the new curriculum and the development of the new curriculum in consultation with CAS.

Speaker 2: Oh, well in consultation with a lot of people. It wasn't just CAS. I mean CAS had a, obviously, had a ... Ultimately there was a consultation group formed that included people from basically, included players from a lot of places including Mace and including Next Gen and including.

Speaker 1: Mace is with an 'M' isn't it?

Speaker 2: Yeah. So, I mean, I think there was an expert group that was the formal body.

Speaker 1: And you've got a couple ... that formal body, publishing a couple of, there's a couple of papers that that formal body put together aren't there? Sort of white papery things that have names like, they've got CAS, and Microsoft Research, and Google and Mile Barry's on a few of them, and I think you might be as well, I don't know for sure but there's these documents.

Speaker 2: So probably the starting point of this was the document I was involved in, which was an attempt to exemplify what a curriculum would mean. What a computing curriculum would mean. That became our first attempt really, to say look, okay, here we are, we're saying that we need it, a subject change to computing, computer science, what does it mean? That was really the first attempt where we identified concepts, abstraction, decomposition, those sort of things.

Speaker 1: Computational thinking.

Speaker 2: Yeah. That was our first attempt to do that and then all the subsequent thing's that sort of grew out of that.

Speaker 1: And, that's one of the ones where it says "computational thinking is a debatable term that nobody answered to." This is what we decided its called.

Speaker 2: Yeah, so I mean, I have all of this stuff in my archive somewhere.

Speaker 1: Well what's really interesting, hearing you tell this story is, I've seen the bits of it, the evidence trail, the Livingstone and Hope Report. Which, I've cited at various times. The shut down and restart report. That CAS paper, and these are things that I've come into and read as an outsider ,as part of this process. But to have that narrative of what's actually going on behind there and as I said, you know, from the beginning, I've had this, well when I started, I was interested in, the impact of essentially industry and governments desire to have computing worker bees, people who thought of computing as something you went and did office jobs with.

 What I've then learned through this process is that, that's completely wrong. For a whole load of reasons. Partly because industry, and I mean industry broadly, so we didn't specific industry people involved, but industry broadly wasn't really involved in this project. There was no consultation with say small computing, software developers to say "what do you need from people coming into your industry?"

Speaker 2: I mean, I think, just again in terms of that landscape, I think that there are problems really with the whole representation of the needs of business in a general sense. So, you have each skills is the other sort of play in this. They are the sort of people who are forever going on about the need for, the workforce to be more IT literate if you like. But again, I just think that their whole history is lamentable, but they come from they had very big influence with the department of BIS. So, again you have two governmental departments really that have different sort of influences. So you have BIS and you have the DFE. So you've got the BIS people.

Speaker 1: Business, Innovation, and sport.

Speaker 2: Yeah.

Speaker 1: You were trying to do, in place training and CPD for people already.

Speaker 2: Yeah and they probably, they have a big history in term of workplace training and re-skilling and all that sort of stuff.

Speaker 1: And they, the European Computing Drivers Licence comes from that side doesn't it?

Speaker 2: I'm not sure because you see, before the ECDL the was CLAiT in schools. You know, there were these mechanical software skills courses. Typing courses and CLAiT was the next step up from a typing course, where you did a bit of spreadsheet work and so I'm very, very mechanical, fickle, etc.

 In a sense, you see there is a separate need for narrow software skills in particular sectors all the time. If, you're going to go with secretarial work you need to touch type. If, you're going to go into accountancy you need some spreadsheet skills. So, there were many vocational qualifications. The vocational qualification sector of education is separate from the secondary education academic side. So you really have the DFE and higher education, and then you have the FE college, surgery colleges, vocational side who are represented through all different bodies and so on. Again, as result there's been endless replication, duplication of things and you end up with this ... The problem you have in the whole vocational world is that you ended up with lots of ... Again, in this privatised world that we're in, you then ended up with loads of commercial companies offering vendor qualifications and dovetailing them with the needs of vocational qualifications so that's how you got the Cisco certifications and your Microsoft Office professionals. I did various endless Microsoft certified professional stuff.

 Part of the problem here was you also have the deregulation of the sector. It was utter chaos, really. That's really one of things that the Royal Society Report pulls out, is that the whole vocational sector, there's just so many different qualifications, actually when you can ask the question what is ICT, it means different things to different people. You have City and Guilds, you have this, you have that, you have the other, and they were all ICT courses.

Speaker 1: And everyone is screaming for whatever they think they need and there's not a huge amount of, because its in lots of peoples interests to claim that every sector has its own skillset. There isn't any kind of sitting down. There isn't any kind of bringing together and saying everybody in general, most people, will need word processing, spreadsheets, da, da, da, da, da. Well if you're going to this you need these-

Speaker 2: Well yeah the problem was, yeah you had all sorts of different players. Again, I remember going out to Cisco many years ago to look at ... because I always felt that for the less abled, more hands on kid, less academic, more hands on kids, there are huge job opportunities in what we call, IT technician work. Cabling, basic IT support, that sort of stuff. So I was really interested in trying to offer those sort of things in school.

 I went down to Cisco for the [inaudible 00:58:50] and again these skills were there. I came away more bewildered than when I went in. There were just so many different qualifications, and they all claim that they tick this box, or that, and this box on that. You needed a degree to fight your way through it.

Speaker 1: And coding or programming within that seemed like a very niche skill.

Speaker 2: Well, you see, within that narrow vocational framework the coding elements were if you like, scripting for a task. I think one of the early things that we were very keen on when we were trying to get the computing curriculum off the ground was to say that "there's more to computer science than programming and there's more to programming than scratch." Those are essentially the two things that we kept saying. Or I think you could say now "there's more to programming than Python, because essentially that's if-

Speaker 1: I was gonna say when you said there was more to programming than scratch, you were halfway there, but more to programming than Python then you're [crosstalk 00:59:59].

Speaker 2: When we were trying to articulate what we meant by computing to most people, if you said "what's the difference between IT and computing", they equated computing with programming. Now whereas right from the early days of CAS, and again if you look at all the back issues that switched on, it was a theme I was trying to push right from the word go was that, actually, if you see ... The problem with the IT curriculum, was that you taught databases you taught spreadsheets, you taught Photoshop, you taught whatever. There was no unifying underlying principle to what you did in Photoshop that you could take into spreadsheets and so on.

 Now, if you take programming as just another isolated skill, that's equally the case. So, it's just another narrow skill. So actually, what we were really about, was saying "what are the unifying concepts that run through, whatever domain you want to explore."

Speaker 1: Well, and that is a really nice segway into bringing this up today, because I think from doing this work, that's one of things that I'm seeing and hearing, is that actually, programming is emphasised in the curriculum. It's one of the few things that is specified. You must be able to programme in two languages one of which is text based. But, it is one of the hardest things for people with little background to teach. It is also, from what I've observed, one of the hardest things to teach broadly.

 Some kids get it, some kids don't. They really learn it at different paces. So, what you end up with, because it kind of becomes the loudest, the squeaky wheel. Its getting the most attention. What we've ended up with, is very much programming particularly python, really. If we're being brutally honest, python possibly scratch, is being taught as a skill in a skills based way without that unifying theory of this is applicable to everything. And a lot of those other kind of broad visions of computing have fallen by the wayside, because its hard to teach python to kids who don't get it. Again, being brutally honest. That's what I'm seeing.

 So, what is the biggest flaw, or the biggest strength, or both of that document that now says Computing Curriculum 2014 on it? What do you think?

Speaker 2: Sorry what?

Speaker 1: What's the biggest flaw with that? What's wrong with it, or what's really wrong with it?

Speaker 2: Well I don't think the document itself is particularly flawed because it doesn't have a huge amount of detail.

Speaker 1: There's not a lot there.

Speaker 2: I mean, in a sense, the flaws come from the fact that there has been no resources or very few resources, geared to then preparing the ground for its introduction. So, yes, there's no [inaudible 01:03:29] emphasis on programming because that's what most people who've have never programmed were scared of first off. They associated it with programming and that's what scared them, they therefore looked for anything to get them off the ground. Python is a low entry etc. But, they never really had the time to develop their knowledge beyond that or to think really, what's the purpose and what it is that they are teaching.

 Again, what you're getting is the problems of trying to drive change in a deregulated education market essentially. Had we done this 15 years ago what you would have done is you'd have got, if this happened in 2000 and there'd been the recognition that we needed to change the subject content, then you would have contacted every LEA and said "who's your IT advisor?" We would have got all the advisors together. You'd have come up with a coherent thing where they would put on courses in their areas etc. You would have worked with the exam boards, hopefully, to make sure they exemplified what you wanted.

 The biggest problem you've got in this was not the document, and rewriting the document wouldn't solve the problem. The problems are that you've got schools that have got little in the way of any wider support. So the local education parties have essentially have gone. The subject area, they're better, which was the government QUANGO is gone. So they're just thrashing about. You've had all sorts of snake oils salesmen rushing in to fill the vacuum with whatever they could make a book out of.

Speaker 1: Worst and best in that?

Speaker 2: Sorry?

Speaker 1: Would you have a worst and best within the snake oil salesmen? Sort of most snake oily and least snake oily?

Speaker 2: Well to be honest, I don't really have any great time for a lot of the people that come into this scene of entrepreneurial opportunity. I really don't. I have met quite a lot of them.

Speaker 1: Yeah I can imagine.

Speaker 2: They're only enthusiastic to the degree that they can cash in on it. But most importantly, what's gone from schools is any training budgets.

Speaker 1: Yeah.

Speaker 2: What we've had is we've had some minimal funding from the Network of Excellence in CAS, which is how we do the regional centres and that sort of stuff. If, when I start a teaching in London, we got an afternoon a week down at what was called the Professional Development Centre, which was an annex of the primary school where we met with the advisor. We met with all the teachers of our subject in the local authority. So we all got to know each other, and we developed stuff. We used to organise things like weekend aways and things like that. You basically had a thriving professional sort of network, community of practise, that was predicated on having an afternoon a week free to go and mix with your colleagues. Now to try and get people out for two half days a year, and given that nobody works ... there's no trying to join the guys timetables or anything.

Speaker 1: Just ask how difficult it's been for the network, for the CAS hub, the HTPs to get people out of school even when its paid for? And there still plenty of people that don't come or can't take up that offer, or whatever.

Speaker 2: And all of this, if you want to exemplify what the subject means, then the only way you're gonna do it ... We've just produced another great subject knowledge tool, and I look at it and I think, it's pretty wordy but actually, people don't wade through this stuff. What they need to do is they need to meet with their colleagues, forge a bit of face to face trust and friendship and work it out between themselves, and debate it and so on. When you look at some of the people in the HTP thing, and you've got Paul Revell, and you've got some real enthusiasts who just get other people thinking, and that's how the curriculum develops.

Speaker 1: And I'm sure this is what, I think this is what you're talking about, is what you see, what I see when I watch the HTP things, as a little of an outsider, I'm becoming more of an insider now but. Is a lot of enthusiast who each have a different strength. Each have a different thing that they've really put a lot of time into. You bring those together and everybody shares that and so the people who are really excited about sorting algorithms say "this is why I'm really excited about sorting algorithms. The people who are really excited about machine codes, say "this is how we teach machine code because it's really exciting." You get that shared enthusiasm and shared skills and everyone goes away with a range of really exciting tools to teach everything.

Speaker 2: And the thing is, its not just the HTPs, CAS Hubs and so on, all of these more informal arenas are places where ,people are given a space to make mistakes. They're non threatening, and you are forging longterm, professional relationships. You can't get that sort of stuff just by CPD that involves a stranger coming up to do a talk, who you don't then don't see for another six months. Most crucially, and this is why I have very little time for what, I call the flotsam and jetsam, is that actually if you genuinely believe that people learn through doing, then actually nobody knows how to teach computing to this age group because it's never been done before. So you've got to do it and then have an arena in, which you can refine your thoughts and reflect on what you're doing.

 So, I mean I think, yes, I look at it now, and I think, Christ Almighty, did I spend years, just to see a bunch of people doing a worksheet on Python. Which, was age inappropriate. Again, to my mind I don't even believe in introducing text based programming until GCSE. I think its too hard.

Speaker 1: Well its distracting It's not that its too hard. I don't think its too hard, at all. I think that primary school kids can do Arduino stuff with C. I think it's distracting. That it takes up a load of time. There's a big difference, and I saw it with you today, it was very informative actually. There's a big difference between thinking ""oh I'm gonna jump in and do my visual basic, my simple basic programme for this game, and writing your flow chart for it first. Then once, if you've got your flow chart, it doesn't matter, which language you're using, you can transfer it between, in fact that's a good exercise.

Speaker 2: Actually if you want a real positive of the emerging curriculum, it's [inaudible 01:11:04] activities. I mean that is something that I think, A, anybody who engaged and begins to see the real value of them, and to my mind I have a visual of a key stage three computing lesson being much more akin to a science lesson, with maybe at most a third of the time actually on the computers. It's all about exercises that develop computational thinking and systematic approaches to problem solving, that's what it's really about.

Speaker 1: So, if we were to say what is the point of the computing curriculum, you would say systematic understanding and problem solving.

Speaker 2: Yeah. So it's developing a way of thinking. That way of thinking embodies not just a systematic approach to problem solving but a degree of resilience and ways to ... This is why I always say "you've got to read code, predict what happens, then observe what happens." When the prediction is challenged by the reality of what you observe, that's actually when real learning takes place, because you've then got to have a mental tool kit that can, systematically investigate the discrepancy between what you observe and what you predicted. So, that's where the toolkit comes in isn't it? It's okay, so how can I now test, how can I debug what I observed.

Speaker 1: And that's where the strength of using code comes from.

Speaker 2: That's where logical deduction and everything else comes in.

Speaker 1: Because, although its hard to write a simple programme and get, actually as a way of learning debugging, which is starting with a simple programme, debugging a simple programme is much easier than debugging a complex programme. Debugging something complicated is quite hard. Starting with something with simple if then statements. So why isn't it saying "this buzz" when it gets to fifteen? Oh it's because my conditionals are wrong. That's quite a useful little technique to teach that thinking process. Even if ... and why not teach them to write it along the way. But that's where the learning should be happening. Not in "oh you've got your parenthesis in the wrong place, or you've got your brackets in the wrong place."

Speaker 2: Which, you know, was again, in the early days when I taught Logo, again in a sense that's why scratch was such a big break through, was that actually you did, you removed syntax barrier. At least you could start looking at things in a much more pattern oriented way.

Speaker 1: So why does it say, that everyone at key stage 3 should have at least one text space line?

Speaker 2: Yeah. You can start looking at these things.

Speaker 1: And you can't get away with saying you don't know, because you-

Speaker 2: Well, but it was ... sorry D–, I didn't realise you were waiting.

Speaker 1: Okay that's fine don't worry about it. I've sent A– the questions. It's because R– talks too much.

Speaker 2: He just keeps asking the questions.

D–: We know.

Speaker 1: If you want, I'd love it if you had a look, but you know, if you're busy with other things don't worry about it. But, if you want, have a look at those questions and see.

D–: And give you a written response?

Speaker 1: Yeah if you want. That'd be fine. That'd be great.

Speaker 2: Well possibly, I mean I'm sorry D– I'd forgotten that you were waiting. I mean if we were at the hub or something like that or whatever. I mean that might ... I'm not sure, but we could always just let B– know that whenever there's something you're maybe going to that [crosstalk 01:14:45]

D–: happy to do both.

Speaker 1: But yeah, feel free. I sent them through because they're, the interviews are quite time gentle anyway. So feel free to have a look at the questions and see if there's any that particularly strike you.

D–: Okay. Will do.

Speaker 1: Anyway, sorry about that and thank you very much.

D–: No problem. It's okay.

Speaker 2: What time is it?

Speaker 1: It is half past three.

Speaker 2: Right, okay. Yeah so what were we-

Speaker 1: What were we saying? Well we were talking about- so why does it say, two languages at least one text space.

Speaker 2: Well so, you see, the final outcome of that national curriculum statement is itself some degree of mangling process, that finally emerged from DfE wasn't really what- [crosstalk 01:15:42]. But equally I'd say that there are amongst the people that were involved in a lot of those discussions, there wasn't necessarily agreement. I would certainly say I didn't like the fact that it made references to text space programming. I think that was a mistake, because I think that was immediately what they've got. I didn't like that it made reference to, knowing, searching, and sorting because that immediately, you could just see immediately how some people would reduce this to some very, very[crosstalk 01:16:15] things.

Speaker 1: I was thinking about searching, sorting. I was thinking about it at the last HTP meeting, because I was realising that sorting, although its not too complicated to explain, if you're teaching it alongside coding or programming, its really complicated to programme. You start looking at the Pythons for a simple sort. It's really quite a complicated thing and if you've got those two things sat next to each other, the assumption is that you'll be able to teach sorting, and then you'll teach it in coding. But that is a huge jump.

Speaker 2: So, yeah. If, you think of this national curriculum came in with that ... I mean it shouldn't have been an assumption but that's what people would take it to be. When, at the same time, the A level computing was introducing the bubble sorting at AS level, as a coding exercise. The problem you have is, I think, it will take some years, for people to work through just what is age appropriate. My feeling is that just as in primary school, in primary school you introduce the notion of an algorithm by getting kids to make you a jam sandwich, or by marching around the playground ,or by whatever, whatever, whatever. Making it at least of how they would get home or. Long before you introduce them to try and code anything.

 It's the same in secondary school, really is that you can understand in unplugged terms a bubble. If I give people a set of cards, and we have a great lesson, and I say "come on, try and work out a way of sorting those." In fact a good exercise I do is a little peg swap puzzle where we've got three pieces on each side. You can do it just as a puzzle and that's a nice cerebral activity. Then yo can say "okay, can you write the instructions for this?" And immediately the people start having to grapple with how do you reference the position, or the context of the position, then introducing an array. But actually you don't get anywhere near trying to code this. Code is the final thing. Actually far too many people start with code.

Speaker 1: Yeah, I've seen it.

Speaker 2: And as a result the experience for children is they spend hours looking for the missing false start.

Speaker 1: And they hate it.

Speaker 2: Yeah, and they hate it. That's what I felt.

Speaker 1: Or maybe, I mean, to be fair, there are also kids who don't find hours looking for the lucky false stop, don't find that level of deep ... but they just find it boring. They find it easy, but they find it boring. They don't see the exciting kind of thing underneath it, they just think computer science is programming, programming is coding, programming is Python, I really don't like Python.

Speaker 2: My worry actually is the prospect of seven years of Python.

Speaker 1: Yeah. Absolutely.

Speaker 2: Again I was very reluctant when they picked up here that they wanted to do Python, Jesus. I said "No. Because, we don't want them doing four years of the same thing." At least if you have change tact with something else at each stage.

Speaker 1: But then you're also not teaching so then there's another part of the story, which I can ask you about because I thought I was going to have to go. The other part of the story is the exam boards. What you have now, is you've got a key stage 3 curriculum, which is pretty vague and has no guidance. So you've got teachers doing the best they can, and then they say "well where do we start?" The simplest question is, which language should I use? We have one text space language, what should it be? So they, look to where do you want them to be at the end of GCSE. And you have the exam board-

Speaker 2: This is the problem at secondary school, is that the exam board ultimately are the arbiters of what gets taught lower down because ... and this was the case with part of what went so wrong with the IT curriculum, but it was also the same in history was, because there were certain things that were important at GCSE. You're teaching lower down the school and teach them again, and again, and again. So in history they did Nazis in three different years in IT. You did databases every fucking year, till you were blue in the face with it. Again, one of the things if you like, is something that annoys me about the way that ... Have you spent a long timeits pushing a snowball up a hill its rather worrying seeing it go down the other side not quite staying in the direction you hoped it would.

 I was involved in the pilot programme for the first GCSE, the OCR GCSE, and it was lovely the way that everyone worked in partnership to create a qualification that everyone was happy with. It was really good. The pilot phase had 200 schools in it. We worked through lots of issues with it. It pioneered lots of controlled assessment and so on. Then, the subsequent revisions, there's no input whatsoever from people, so you've actually just had imposed on you one of the biggest problems you've now got. Again, it's the way governments get people to dance to their tune. The soft censorship that goes on, but the exam boards have just taken this government mantra of increasing rigour, and have revamped their specs with the most unrealistic expectations at every level. So you've now have got really hard qualifications. [crosstalk 01:22:07] And so because they're so hard it's pushing down, and as a result you've got people then introducing stuff at a lower, and lower age.

Speaker 1: So, for example the bubble sort, which was a AS level is now all the way down at year 7 for some people I would think.

Speaker 2: Yes, which is ludicrous.

Female speaker: You're not going to have time to do D– are you?

Speaker 1: D– already left.

Female speaker: Oh yeah, yeah, I am next, are you gonna ... I'm in a meeting two doors down in RA2.

Speaker 1: Right.

Female speaker: Yeah. So I don't know how you-

Speaker 1: [crosstalk 01:22:32] statement.

Speaker 2: We'll try and wrap it up, I've got loads of marking to do. I've got-

Female speaker: So are you going to ... I'll say good bye now.

Speaker 1: I'll see you later.

Female speaker: I've been [inaudible 01:22:41] I'm all cold and wet.

Speaker 1: That's all right. I'll show B– out.

Female speaker: Yeah. Yeah. Yeah. Yeah.

Speaker 1: Make sure he signs out and everything.

Female speaker: I might keep you up on that-

Speaker 1: Yep I'll be happy to ... shrimping is one of the things I'm actually quite happy to deliver.

Female speaker: Yeah. Yeah. Yeah. I was thinking my year 10s could come in and just do one, of Shrimp packs.

Speaker 1: Shrimp pack we will. See I normally, in most of the schools I do this sort of stuff in, I volunteer to come in and do a lesson, except to weeks ago when I was her, I realised that there is almost nothing that I can teach, that you guys can't teach better than could. Because, the [inaudible 01:23:15] is so broad that everything is covered. So I said I'd come in and volunteer, but I can just help out. I'll just watch. Shrimping is something I can-

Female speaker: They like to see another face, don't they? I'm sure there's lots of things-

Speaker 2: Shrimping It, be good. That would be good. Yeah that would be good. Good.

Female speaker: My year 10s can do, because they don't listen [inaudible 01:23:35] computer, they talk about the components and the CPU, [inaudible 01:23:37]

Speaker 1: Well my next personal project, in fact I just ordered 50 transistors this morning, because I want to make a binary computer out of transistors. This is my goal to see if I can do it. Binary [inaudible 01:23:50]. Beyond all of this, I think that's the magic. When you can say this chip it does a load of stuff, but this is what's-

Speaker 2: Actually, just to get onto your list of things that I think is a bit of a concern is, is that because coming from a D and T background, ultimately the power of making is really, has huge potential. But, it's the smoking barrels of our subject in a sense. Isn't it.

Speaker 1: Mm-hmm (affirmative).

Speaker 2: I think that an over emphasis on trying to promote that sort of stuff is really off putting for a lot of new teachers. Because it's, again there's a lot of base knowledge ... If, you've got 30 kids ... I used to teach electronics and D and T. It all fell off the D and T curriculum for precisely the same reason that, it's damn difficult to manage in a classroom environment. So, these sort of things represent the pinnacle of what you maybe want to, but actually they're not very easy to deliver. I worry about the [crosstalk 01:25:07] but actually it's ... I worry about the fact that the most PR seductive stuff that you can do, get something like VEX Robotics that looks good in school publicity and so on. But, actually its, we need to try to popularise that, like you say just the basic unplugged stuff, that they-

Speaker 1: Well there's the basic unplugged stuff and there's the basic computer science stuff, but the other part that I think is really, again this is one of those presumptions that I came into this research thinking I had the answers to it, and I've learned I really don't. There is a visioning of the future. Part of creating a curriculum is a visioning of the future, and trying to anticipate. Because the nature of education is you're trying to anticipate what skills, in some cases a five year old, is going to need in 20 years time. That's really difficult.

 Part of that process is to say "what do I think the future will look like in 20 years time?" For example, I think the Vex Robotics stuff is interesting within that context because actually, there's a decent amount of research from people who've looked into this enough that, robots and artificial intelligence, but physical robots are going to play a fairly important role in particularly manual jobs. If you want to go into a manual job, be able to operate a robot, and having built your robot is probably an important skill. It's probably a key skill.

Speaker 2: Yes. Again, I think one of the problems if you're working with this envisioning the future is that actually, what is required, and I think there's a genuine recognition really amongst a lot of educationalists, is that what is required in terms of empowering people for the flexibilities of an uncertain future is absolutely at odds with the structures that are imposed in education now. That's the problem. Actually that's what can potentially can get quite frustrating is that, when I've spent my entire time in education listening to visionary's talking a lot of sense, and knowing I haven't the slightest hope in hell of influencing that.

 A good example I think is the raspberry pie. The raspberry pie it's spawned this huge maker movement, it's tapped into the maker movement. There is a massive maker movement, but actually that whole maker movement stuff it can only work in ... Ultimately if you want proper education, to educate people for tomorrow then you make schools cross-curricular, project based environments. Therefore, you can't have rigorous assessments, or rather narrow assessments the like of, which we have. That's the problem isn't it.

 At the moment the assessment tail wags the dog. That assessment tail is so narrow that actually, you end up with people essentially teaching to the test, and losing sight. As a result you get children not getting any knowledge, they just get the shallow polish to get over the next hurdle.

Speaker 1: There was a great little Radio 4 thing the other day about teaching for the test and how damaging it is, to people who then become professionals who deliver to the benchmark and never have creativity. The people who do have creativity never are successful. You can't be successful if you're creative and taught for the test.

Speaker 2: I've always, I feel there's a general recognition of that sort of thing in education, but ultimately you are bound by the pressures you're under. I think what you see amongst people that are in teaching, good teachers know how to-

Speaker 1: Teach well.

Speaker 2: Yeah. How to keep rooted on trying to grow knowledge and have the confidence to know that actually, if you keep tilling the soil then you provide a rich environment. Then the growth will happen. That's very, very difficult. It's quite easy here. It's very difficult, like when I taught in London we're you're under the cloche, and you're on special measures if you don't get this number of people through, you really don't have the professional freedom to follow your own judgement . I'm quite fortunate to be able to do here. Partly by virtue of the fact that we've got really good children, and well disciplined [crosstalk 01:30:12]

Speaker 1: You have more infrastructure. You're in a good-

Speaker 2: And, I've been teaching for a long time. I've got the confidence to know that I can ride it out and-

Speaker 1: Here's the part of that, that I think, there's two parts, there's loads that I want to take out of that. I can't do it all at once. I mean part of it is one of the things I find so encouraging about interviewing teachers particularly, is a lot of parents look at secondary school, and they feel so depressed that it's just preparing their kid to be a cog in a workforce. What I find is, whether or not the curriculum is trying to that you meet amazing teachers, and amazing kids who really are resisting that. Either actively or just passively in a "No. That's not what I'm doing. I'm growing knowledge." That is really, incredibly encouraging.

 The other part of that is this idea of project based learning, which again I think is, I'm Chair of governance at [inaudible 01:31:11] School, and they emphasise child initiated learning up to year 6. I keep saying to our head "this is the only system of teaching that makes sense for the 21st century." ICT and computing seem to be a vital part of that, because in order to be a good project-based learner you need to be able to evaluate information. You need to be able to access tools that make you more efficient, like computer tools. You need to be able to understand how those tools work.

 ICT and computing seem to be one of the few vital skills that if you don't know them, and if you don't have them you'll struggle. Whereas something like Science, fact based things as they develop you need those ICT skills in order to access them.

Speaker 2: Absolutely. Yeah. It is increasingly that sort of underpinning discipline that's-

Speaker 1: And underpinning in a way that English is underpinning. In the way that mathematics is underpinning. That if you don't have access to these skills you won't be able to access a huge aspect of the world. And you can-

Speaker 2: Your ability, essentially your ability to be a life long independent learner. That's essentially what it ... yeah, that's what all this is predicated on, isn't it. Is that actually, in the future people are not going to be in a job for life. They're not going to be in a trade for life. And they're actually such is the technological change-

Speaker 1: No. No. That's already happening.

Speaker 2: Ultimately what you're trying to do is develop people with the capacity to retrain themselves on a regular basis.

Speaker 1: Yeah, and you need, and I think this is one of those interesting splits between that intensive detailed ICT curriculum and this broad, vague computing curriculum, is actually the skills to do that are somewhere in the middle. Although it's useful to know how a computer works, you do need a degree of ICT skills to be able to retrain yourself using computing technology.

 The other part of that I think is well-

Speaker 2: Well again, that just there is an interesting aside isn't it? It's my observation of, this obsession with data driven schools and measuring progress is that what I'm finding, just is that I think everyone is finding there's a bit of an overwhelming proponents of [inaudible 01:33:54]. I think you were also getting, when you look at what we might call the more broader computer science stuff there's this curious divot, this false divide between programming and theory. The theory itself gets reduced down to what can easily be measured. You do, you get this ... How a computer works is utterly irrelevant almost, but it's a very nice easy thing that you can draw to them, you can learn three terms, and so on, so it becomes something that's easily testable. If, you really stereotype the worst of the computing syllabus you end up with people who basically a load of binary. Let's do binary. Loads of binary conversions because you can test that. Some disparate and unconnected facts that constitute the theory and programming and some narrow skill of programming. That's it at it's worst, or what I could be at it's worst.

Speaker 1: I worry though having done this work for the last couple of years is, that actually is a better description of what's actually happening in the majority of schools than anything else, is exactly those three things There's a load of binary conversions, a bit of disparate facts, and some programming. That's what the computing curriculum has become.

Speaker 2: I think it is early days.

Speaker 1: Yeah.

Speaker 2: You see to be honest, if we say that's what's its become at now, and I think that's what it started off as, really in many respects. When I think back to what the state of IT was like, when that was introduced in the national curriculum, where nobody had clue. When, you had your English teachers getting threes lines of processed text of out of someone. It was just as big a mess then. The difference was that you actually had, like I say this infrastructure out of, which things developed. People clarified their ideas.

 I think the big problem we've got here is that there is outside of the orbit of CAS where there is that sort of professional dialogue and that community practise is that there's very little to challenge, to get people to reflect on this and move it forward. That's why I hate a lot of this commercial stuff. Commercial text books and that sort of stuff. I just think they're diabolical. They help with this reductionism, or they give it all this legitimacy.

Speaker 1: Yeah they do. They become an authority that teaches that as opposed to.

 CAS is limited, and I think that is over abundantly clear from my experience before, but also from the story of CAS. This little group that really started as a forum of 350 people.

Speaker 2: CAS was started as a dozen people.

Speaker 1: Or as a dozen people. It's amazingly big, and amazingly good at certain things, and there's another conversation about what it's not good at. This isn't even a criticism it just can't reach every school. It's not setup to be able to deliver CPD or support to every single school in the country.

Speaker 2: No, but I think it could be.

Speaker 1: I don't mean that even as a criticism either.

Speaker 2: Actually I think the biggest problem is, it isn't thinking bigger. I think it needs, the lobbying side needs to be saying "this is heading for disaster if you as government do not reinvest funding require schools to" ... The model for me is not, in fact again I worry about CAS developing to something where we get people who are in paid employ who are not in schools. I'm not really interested in a bunch of [inaudible 01:38:16] who are, who almost become another set of flotsam and jetsam in the teaching profession. What I want is, I want CAS to lobby the government to reinvest funding to replicate that idea of releasing teachers for an afternoon a week. If you can do that in every locality then, actually CAS could grow. We've got the local infrastructure on, which that could really flourish. That's where it-

Speaker 1: Yeah, so if build on those CAS hubs, which are being grassroots driven and start by releasing the hub lead, and then release someone else so that-

Speaker 2: But you don't want to release them, you see we don't to give-

Speaker 1: I mean for an afternoon.

Speaker 2: Yeah. Oh yes. Yeah.

Speaker 1: Release them for an afternoon a week. You want to keep them in classroom you just want to make sure the funding so that they can leave the classroom once a week.

Speaker 2: Yeah. And have that space to think.

Speaker 1: Well they need each other. I think, and this is something again a broader issue in teaching is having a meeting with other teachers that isn't at 6pm. But is at 2pm-

Speaker 2: In the afternoon, it's leisurely.

Speaker 1: So, that you're still going home at five.

Speaker 2: At a normal time.

Speaker 1: At a reasonable time as opposed to, in order to attend the CAS meeting, CAS hub meeting, or governors meeting, or a whatever else you have one more thing at five or six in the afternoon that will take to six or seven in the evening.

Speaker 2: You see I think the problem with all previous ideas of, ways you change the curriculum were that they were essentially top down. You go advisors in, you got this in, you put on CPDs as so, but actually if you have a vision of a model that can grow from the grassroots up, there are all sorts of ways you could perhaps make that, provided you got some buy-in to fund this, there are all sorts of ways that you could tie that funding to a requirement for teachers to show willingness and so on. They could give a bit of life. Having lived through the IT curriculum it was profoundly disheartening to see lots of good teachers leave the classroom to become project managers. And that you could see the impact that had on them as well, and the fact that they ceased to engage with the problems that the teachers were having. Instead, spouted the[inaudible 01:40:48]-

Speaker 1: I think one of the most interesting things you said about having that time, was having the support and infrastructure to takes risks. That if we look at the way that computing is being delivered, it's being delivered in a very risk averse way. We can't be too critical of that because there's lots of reasons that make that the only way it can be delivered. So, if you can learn Python in a summer, you learn Python, you do Python, and you don't need to know it brilliantly to be able to keep the kids going at reasonable level.

 You don't think that maybe next summer I should go and learn some Java, or you don't think because nobody has done the digging to find out. What is the most useful language? You don't even think, why should I teach Python? Why is Python a useful thing to teach? You think I need to teach a text space language, so I'll teach Python. They'll learn a text space language and I'll have done what I need to do, to do that tick. There is something about, is Python the best thing to learn, or what language would give kids the best overview to be able to read lots of different languages? But, there isn't the time to do that.

Speaker 2: That's what, really building a community of practise is all about then. There's plenty of research about what leads to successful professional development. At its heart really is that idea of a self sustaining community of practise. That has a shared trust amongst its members.

Speaker 1: To bring this whole thing into land, which is we are where we are. Is it a step forward? Is it a step back? How would you assess where we are now compared to the history lesson you've given me today?

Speaker 2: Well it's definitely a step forward, and a significant one. It's repositioning.

Speaker 1: Is it a step in the right direction?

Speaker 2: Yeah, it's a step in the right direction. It is repositioning computer science as an underpinning discipline, academic discipline.

Speaker 1: Okay.

Speaker 2: But, so in the 25 year view of things, I think we're picking up and rediscovering many of the early aim of the early pioneers of computing in schools. From the '80s onwards and the [inaudible 01:43:54] that person who saw computing as a tool for thought, for developing thinking. It certainly a significant step forward compared with the dire state we were in when everyone was doing for unit GNVQ. When the subject had effectively, was so devalued that nobody took it seriously. I think potentially in a sense, you've gone through the great rush of building a rocket.

 Getting in on the launchpad. Getting it up in the air. Whether it's going to explode before it reaches, gets out of the atmosphere is another matter. To be honest I look it and think unless there is something that can come in, in terms of resourcing that will allow us to significantly scale up the principles on, which CAS has been based it could implode.

 But, when I say something what worries me is, in the absence of funding that is of, sustained funding that is protected and gets to the right place you will end up with ad hoc initiatives that are often, the education imperative is subsumed by short term needs to satisfy the, usually the corporate sponsor that provides the funding. That's what bothers me is that I think there's a lot of frenetic energy to try and do stuff with whatever money you can get hold of.

Speaker 1: And that seems, in computing I think even it's not even like you've got a whole bunch of corporate sponsors, and coming in and funding things. This is the conclusion that I'm coming to more, and more is that, that the computing curriculum as it's being experienced by the kids, which the thing I'm most interested in, in some ways. What is their experience of it? Is that it's not meeting anybody's needs. It's not meeting the corporate sponsors needs because there aren't any corporate sponsors and if there are, they're not being clear about what they want.

 It's not meeting the kids needs because they are getting the amalgamation, and they aren't sure what it is, and they aren't sure how it's changing their future. They're really confused about what's happening in terms of their futures. They see computers as important. They see understanding them as important, but they don't see the connexion between what they're doing in the classroom and that thing.

 It doesn't really meet teachers needs because the time goes into these complicated hard things, which is really Python. Been really specifically to name it.

Speaker 2: But, depends where you are doesn't it. I can point to plenty of people that I know who teach it, and teach it well. I mean I look at it, it goes in and out, 200 million for music in schools [inaudible 01:47:13], and I thought hang on, why haven't we got 200 million? I was over at York at the Stem Centre and I'm thinking this is the sort of thing-

Speaker 1: Stem is getting? I can't understand why stem ... In fact a friend of mine who helps us with our code club but also works for a developer company. Great kid. Really excited about computing. He's just trained as a stem ambassador, so he can help us with our code club actually. He said he did computer science, and they were just so excited that there was somebody. Stems one of those structures where you think, why isn't computer science at the heart of that because it can magnify all of the stem.

Speaker 2: Yeah. I know.

Speaker 1: It's not even that it's a stem subject. It magnifies all of them.

Speaker 2: Yeah.

Speaker 1: Every single stem cell checked.

Speaker 2: The Royal Society report was top ended by Sir Paul Nurse, biologist. I think there is a recognition inside ... these things are changing. I think there's a recognition inside science that science teaching itself needs to change. I think there's a recognition in math and all sorts of people, that math teaching needs to change. If you look at Wolfram mathematics and stuff like that-

Speaker 1: His talk about how calculation is a irrelevant skill.

Speaker 2: Again, you see this all the time is you have very, very insightful people, like people like the Wolframs, like people at the top end of the stem. Who'll see how their educational visions forever keep crashing up against these structures that are imposed in school education.

Speaker 1: Yeah. Yeah, absolutely. Or you have people like, not [inaudible 01:49:16] like Khan Academy, which is trying to do something different because it doesn't see the system worth fighting. There creates an alternative.

 Final question, and I think I know the answer, and a simple answer is fine. If had a magic wand to give you, and you could wave your magic wand over this whole thing, if you had your one wish or if I'm generous three, what would you change?

Speaker 2: Well the immediate demand for me would be funding to release people to build a community in working hours.

Speaker 1: Yep.

Speaker 2: That's the immediate thing. I think on the broader thing, league tables are the killer. It's the league tables that kill education. That on the policy side and a return to the idea ... It's not just league, league tables are at the top end of the pyramid of monitoring, that stultifies teaching. Ultimately you need to, the long term aim is to re-empower teachers as innovative professionals. It's [inaudible 01:50:44] to me-

Speaker 1: Because you are.

Speaker 2: It's [inaudible 01:50:46] to me that, I never look at other materials. If I need to teach something I learn it myself and think about the way that I might want to teach it. I might be able to make it interesting. It's an absolute [inaudible 01:51:07] to take something and deliver it.

Speaker 1: Now specifically about computing either in the delivery or the curriculum, what would you change?

Speaker 2: Specifically about computing?

Speaker 1: Computing. Either in the curriculum itself as document or as an idea, or in the delivery? So, because the community thing is kind of part of the infrastructure, and I agree in that it would make a huge difference and probably is the best, it's the best practical step. But, if we can skip the practical steps, what would you like to change about what's written in the document or what would you change about how people are teaching it?

Speaker 2: The immediate thing would be to try ... It's about exemplifying that there is more to computing than programming, is the crucial thing. Ultimately that the subject is about engaging in techniques for solving problems. That's where programming has it's ... it's impact is in the practical expression of problem solving, but it's the problem solving itself that gives the subject it's power. That's what gives it, its educational rationale and that's what is transferrable to all these [inaudible 01:52:33].

Speaker 1: That's brilliant. That okay.