# School A Teacher 1 – 7 July 2016

Speaker 1: Yes, she's pretty cool, my wife is, yeah she's great, isn't she? And then, this morning on the radio, they completely slighted her because someone's likely checked her references and none of it's true.

Speaker 2: None of it's true, yeah.

Speaker 1: She's not been in charge of big teams, she's not had big financial funds to deal with. She's just pretending she did.

Speaker 2: Yeah well.

Speaker 1: Not pretending but saying she did.

Speaker 2: Yeah, and it's not even just that they've gone back and checked but, peoples that she said she worked for have come forward and be like, well.

Speaker 1: Yup, yup, yeah, you heard all that, you heard all that.

Speaker 2: Yeah, it wasn't quite like she said it was.

Speaker 1: Which is a shame, because again it ruins the impact that she had the other day, which is probably quite positive for most people. So yeah, Theresa May it is.

Speaker 2: Yeah well.

Speaker 1: It's not gonna go is it, for goodness’ sake.

Speaker 2: Not at this point.

Speaker 1: Although I reckon that someone should be on a negotiating team perhaps, I don't know

Speaker 2: Maybe.

Speaker 1: That's her problem.

Speaker 2: Go overs.

Speaker 1: That's going to be Theresa May's problem, is finding a team to negotiate.

Speaker 2: Yeah.

Speaker 1: She wants somebody from Cross Pies as well.

Speaker 2: The whole negotiation process is going to be. I don't know it's-

Speaker 1: Be a nightmare.

Speaker 2: Yeah, it's going to night- and it's not going to take two years. It's going to take years and years and years.

Speaker 1: Yeah, and while the uncertainty remains.

Speaker 2: Yeah.

Speaker 1: We fall. Right.

Speaker 2: Who is the seventh of the seventh?

Speaker 1: I'll answer your question. No, I haven't signed the sheet.

Speaker 2: That's fine. First question.

Speaker 1: Nope.

Speaker 2: N– said the same thing, or P– said the same thing.

Speaker 1: I will obviously.

Speaker 2: Yeah. I'm sure you will.

Speaker 1: Should I sign a sheet now, if you want?

Speaker 2: No, of course I don't have one. That would have been the smart thing to do. So I get-

Speaker 1: So, you can sign the sheet now but go on airfield.

Speaker 2: This is something that I've been realising about the difference between working with kids and working grownups and doing- coz my supervisors have done work in businesses but consent is a lot easier. Coz you take it in and you say, "Oh, do you give consent. Here's the sheet. Sign it." And with schools it's they send it home, wait, remind them.

Speaker 1: I was quite impressed that you only one that said, "No."

Speaker 2: One said, "No."

Speaker 1: When we've got a full list of the rest of the students then.

Speaker 2: Yeah.

Speaker 1: Wow. That's pretty impressive.

Speaker 2: One that said, "No," completely. There's some that is hasn't come back yet or we don't have it.

Speaker 1: All right. Okay. How many of those were then doing it? A few?

Speaker 2: I think possibly up to a quarter. Maybe a few more.

Speaker 1: Okay.

Speaker 2: There's the one lesson that I've not been in and I don't know how many of the ones that we don't have back are from that lesson.

Speaker 1: All right.

Speaker 2: I need to split them up into sets and then I can figure that out a little bit more. But they have had a chance to say no.

Speaker 1: Should have written the set on the top. Shit. I'm really sorry. I didn't think.

Speaker 2: That's all right. It's not your fault.

Speaker 1: Right. Oh well, that's better than I thought. I was expecting you to say there's none. Everybody says no. And there's none come back.

Speaker 2: No, I mean there's-

Speaker 1: But if they're not prepared to send them back in.

Speaker 2: Yeah, exactly. They clearly don't feel- they have the chance to say no and if they feel strongly about it they'll send it back in and say no.

Speaker 1: Yeah, if they haven't said no then.

Speaker 2: Especially knowing what it's like being a parent.

Speaker 1: Are you able to assume if they haven't said no.

Speaker 2: I think I'm able to assume that they don't object strongly. So then-

Speaker 1: But you can't necessarily use any information about it.

Speaker 2: I can't use- but I'm finding that I'm not writing down that much detail of the kids anyways because the classes are so big and because I don't know their names. At the moment it's just, if I take a picture of their work or them I ask them first and if they said no then I would probably find.

Speaker 1: Yup. So you get it sent back.

Speaker 2: And then do a lot of pictures of things like whiteboards, which you can't object to really. No, I think it's fine. Clearly they don't object strongly and then also there's some people who have said, "No photographs." Some people have the check marketers clearly and take them and some people have not and then some people have said, "No, we'd didn't have a chance to ask questions." You think but you could have done.

Speaker 1: Or they did. You introduced yourself plenty of times, didn't you?

Speaker 2: Yeah. It's-

Speaker 1: Especially if they hadn't returned it at that stage.

Speaker 2: Yeah. I mean, I need to get better.

Speaker 1: And they were offered to email you or talk to me, so.

Speaker 2: Yeah, exactly. I think if I were to do it again, I would go back and I would change how that's written slightly so that it is- instead of saying, "I've been given the chance," "I understand that I will be given the chance to ask questions." And then-

Speaker 1: So we were you first, kind of, school?

Speaker 2: Yeah. For this sort. I've done Key Stage one stuff before but you don't ask the kids for Key Stage one. You just-

Speaker 1: Ask the parents.

Speaker 2: Ask the parents.

Speaker 1: Yeah.

Speaker 2: You just say-

Speaker 1: As much-

Speaker 2: And key stage one stuff in very small schools where they see the parents. It's super easy because-

Speaker 1: You just wait at the gate.

Speaker 2: Just wait at the gate and like, "Do you have your form back?" Well yeah, sure. Or they say, "No." And they say, "Okay, here's one."

Speaker 1: Sign it now.

Speaker 2: Yeah, so. It is different. It is a learning process.

Speaker 1: Right go on then.

Speaker 2: Questions. Have you looked at them at all, by the way?

Speaker 1: Yes.

Speaker 2: Yes, fine.

Speaker 1: Do I have an answer to them all? Not really.

Speaker 2: No. That's fine. I mean, you don't have to.

Speaker 1: Because if I did I wouldn't probably give you the answer that was- yeah.

Speaker 2: I mean, it's a summer structured interview. So they are ideas more than questions and just makes me look more regressive. I've written the questions down, haven't I?

Speaker 1: No, that's fine because it means I can, yeah.

Speaker 2: Starting off. The first ones that I said are kind of about your interpretation of the computer curriculum. The other ones are about how you deliver it, evaluation and impact.

Speaker 1: Yeah.

Speaker 2: Does that all make sense?

Speaker 1: Yup.

Speaker 2: And the other thing to say is, feel free to go off topic if you feel like there's something that I'm not asking the right question for but is the right answer. That's fine.

Speaker 1: Okay. I'm good on that. It's fine I'll [crosstalk 00:05:49]

Speaker 2: How do you interpret computing curriculum or computational thinking?

Speaker 1: Well, I really I think it's the ability to solve problems, full stop. We could break that down into- we could decompose that into various parts but I think if you wanted it, for me in a nutshell, that would be it. So the ability to do that with a view, I suppose, to writing an algorithm to describe a solution that problem.

Speaker 2: Okay.

Speaker 1: I know we spent quite a lot of time at the first aid GP trying to find what computational thinking was. Or a computational approach but ultimately we spoke about abstracts and decomposition algorithms, part, and recognition. Things like that. So in reality that was tools to solve a problem, I would say and as I say, a method to describe the solution to that problem.

Speaker 2: Do you feel like the computational bit is- of the computational thinking, how important is the computational aspect of that?

Speaker 1: You mean it's linked with providing that solution for ultimately writing a programme so to speak?

Speaker 2: Yeah.

Speaker 1: Well I don't necessarily think it is. I do think it's a skill, its approach. Isn't it, really? You could argue some people are born with the ability to do that for a problem, to develop a solution. Does that mean they're born with the ability to computationally think? I don't think it does. I think they've just got that. I guess computational thinking arguable is perhaps teaching methodology for solving problems and that's just a name that's been given because it's most specifically with solving problems or because of the algorithm part with solving problems for computer, I suppose, for programmers to solve problems, I guess. If that makes sense.

Speaker 2: Yeah, no, it makes a lot of sense.

Speaker 1: As interestingly enough, I had a boy yesterday who had come with coding clip, he said something. He said, "Um, when are we gonna stop all this thinking coz I just wanna do some coding." And I had to justify it by showing him this, which is one of my recent acquisitions. And the first chapter is computational thinking.

Speaker 2: Computational thinking, yeah.

Speaker 1: So, I don't tell you, well they define that as. I'm sure they've got a definition in here somewhere but, yeah. I said to him, "So, look at the end of the day, maybe by the time you get to year 10 we won't need to worry so much about that, for you."

Speaker 2: The computational thinking.

Speaker 1: So here we got, computational thinking involves playing a set of problem solving skills and techniques that are used by computer programmers to write programmes. I guess I would say that, yeah. All right. So it's the planning stage of the system's lifestyle, isn't it really? Well actually when you decompose it, that's the analysis and perhaps the success criteria, to a degree, what you trying to achieve.

Speaker 2: The planning stage of the system's lifestyle.

Speaker 1: Yeah, of the system's lifestyle you can argue, but even then-

Speaker 2: Do you feel like its something that could be taught?

Speaker 1: Yeah, I do think it can be, yeah. I think, clearly like everything else, students will find it easier than others but we're arguing that this is a computer science than the methodology of science can be taught. So therefore, I would say that you could teach them. At least attempt to do it that way. It may not result in doing that. I mean, I guess the way to test it would be to- I've got a book up here. It's all about- Is it that one? No, it can't be that one, it must be this one. It's just full of- No, it's not that one either. It's just full of- It's probably at the back. Of loads of algorithms and stuff. An attempt to- I should read that, it's quite good. An attempt to- It looked good but revolting. An attempt to solve this and there's loads of algorithms that cause problems. Whether that then would then work.

Speaker 2: Yeah.

Speaker 1: We don't know. It would be interesting to see, to apply it to these random, kind of, challenges, which I'm about to do actually with my computing group.

Speaker 2: Yeah. [crosstalk 00:09:48]

Speaker 1: Apply everything we've done so far.

Speaker 2: Yeah.

Speaker 1: To a challenge that will require them to, kind of-

Speaker 2: Well that's sort of where we're up to with the main group.

Speaker 1: Yeah, but I've overtly taught them decomposition, part recognition, algorithms and Sudo Code, and trace helpers, and dry run testing. All of those bits that they should then get a need to apply to it's one challenge. So it's slightly different, I think, to what we'll be doing next because we haven't overtly taught those students that way of thinking.

Speaker 2: Yeah. So, is it in computing club? Or a-

Speaker 1: No, this is my GCSE. You get ten group, the guys that are doing this because of course their GCSE's split into two papers. Computational thinking and data representation. I don't know why they've put that in there. Should have just left it in the other. Probably to make it two hours, I suppose, two and half hours and the other ones to the rest of the theory. Don't want to just keep you on one paper, it would have been better, but.

Speaker 2: And they have to write, sort of, an essay about that?

Speaker 1: No. They actually have to apply their skills.

Speaker 2: Are they a given a problem?

Speaker 1: Yeah. No, a number of problems as far as I can see but whether it's Sudo Code or flowchart, or just a text based, kind of, description and algorithm, I suppose. And then they have to prove things. Again Max, want me to fix this science and therefore, we're trying to prove things. I mean one of the best things that we could perhaps be teaching our Key Stage 3, which we're not yet, is proving an algorithm by using a trace table or dry run testing it. Why we're not doing that? Because we haven't had to do it until the new one came along. Now we're having to do it at Key Stage 4. We should probably be looking to do that at Key Stage 3. So that gets them to method-

Speaker 2: Methodologically. Methodically.

Speaker 1: Yes, thank you. Think through methodically. That's-

Speaker 2: That's the word we're looking for.

Speaker 1: Yeah. That's why I was not getting there. Requires them to methodically think through the problem and prove whether or not their solution is actually going to work. And again, I think that will be integral to the computational thinking. So not just coming up with a solution but actually understanding the testing of it.

Speaker 2: Do you feel like, I mean this is sort of question but also a broader question. When you're using computational thinking to a solve a problem, do you feel like you tend to come up with same sorts of solutions?

Speaker 1: Interesting. I suppose we should be doing really because there should be patterns that are reusable. That the skills or methods that I've applied elsewhere that I should be able to reuse that. I suppose it should be but I don't really think. In fact, in all honestly, I would solve a problem and then forget how I solved it. So all my students in my Key Stage 4 year, we having class will be going, "Ah, so how does it work?" I'm looking, I'm thinking I can't remember. I have to go back and have another look. It just happens. I lug it and then move on, which is not a great way to work really, I suppose.

Speaker 2: I'm the type of person where I like little puzzle games but I have this terrible tendency to fiddle the solution. I find the solution and then I think-

Speaker 1: Oh that's how I did it.

Speaker 2: Or even, Oh, I guess that works. Now what did I do.

Speaker 1: I wold have done that approach and I probably still use that approach myself and then I might refine it and might go, actually that probably be quicker if I did it that way.

Speaker 2: Yeah.

Speaker 1: I suppose I'm getting a more generic approach that's, it's like, you know, it might have a little efficiency that could be built into it but otherwise it's just the principle, isn't it. But I haven't practised what I preach, much. I should do, probably. But when I do that, yeah, actually it is a lot easier to solve the problem.

Speaker 2: It's probably applying the contradictional thinking steps.

Speaker 1: When I do that, it does work a lot easier.

Speaker 2: Yeah.

Speaker 1: It does solve itself quicker. We had a problem that we had to generate a solution to a mean average without division.

Speaker 2: Okay.

Speaker 1: So practically, there is a way to do and then- because this LMC, you know, like that bell tanks thing that we did at- Did you see the bell tanks at the conference?

Speaker 2: Yes, I did.

Speaker 1: So based on that language, the computer language.

Speaker 2: The source code, sort of?

Speaker 1: Yeah. Very limited. Eleven machine codes.

Speaker 2: Machine codes.

Speaker 1: So, eleven instructions. None of which included multiply or divide.

Speaker 2: Okay.

Speaker 1: And try and work out the average, which is actually, and I solved it on a whiteboard because I actually did do a trace table. I didn't think about what I was doing. I just actually wrote it out and went, oh right, okay, that was the solution. Went home and dreamt the solution, woke up, typed up the solution. It worked. It doesn't always happen but I guess of thinking about the problem in that way. Thinking about computational is going to, kind of, provide that kind of leverage, I think, for coming up with an answer.

Speaker 2: But it's interesting because there's a weird mixture of ways you're describing it, which is fine, of both quite a creative process. There's process of do a bit of work on it, go home, leave it, dream a solution but that process of leaving and kind of allowing your mind to work on the problem, which isn't necessarily how we think of a computer science methodology. It's not the kind of step by step.

Speaker 1: Thrashing it out.

Speaker 2: Thrashing it out war card. Decompose it, write out but a far more creative process.

Speaker 1: Must be the musician in me.

Speaker 2: Mmm?

Speaker 1: Must be the musician in me.

Speaker 2: Yeah. Well no, but I think it's actually a- I wouldn't even call it a contradiction but it's an aspect of computing that is part of what exploring, is that computing can be quite a creative process.

Speaker 1: Repeated eureka moments.

Speaker 2: Yeah. I mean and if you talked about developers in industry, they describe that but it also can be taught as or sometimes understood as being quite a analytical process.

Speaker 1: Yeah. I mean the arguments there that you could argue perhaps, that perhaps- And I don't mean to sound horribly, but you've got programmers that do just programming and then you've got those people that are generating the eureka moments that are passing on the requirements for the programmers to programme, I suppose.

Speaker 2: Right. And it's interesting to look at how do we teach it, how do we frame computing within school. I mean, do we frame it as an art, where from as a science, do we frame it as something, as something a bit like cooking where it's a mix of both?

Speaker 1: So that makes it a humanity, right?

Speaker 2: Right. So what is this work you do?

Speaker 1: So we should be getting a BA not a BSC in computer science. Well I guess I can see where you going from there. Yeah. It's the same issue with economics, isn't it really? It's a mix of science and bullshit. Just for the record. Fans and bullshit. Yeah.

Speaker 2: And guess work in economics.

Speaker 1: Yeah. I've not quite thought of it like that. I've not really sat and analysed how I solve stuff.

Speaker 2: No, no and-

Speaker 1: But I do have moments. You know, it's the same with some oke said somewhere or something like that and I don't keep a note pad next to my bed and I should do really because sometimes I just try and hold onto it until the morning. It's like needing the loo and not going and then when you get there you don't need to go. If that makes sense.

Speaker 2: Yup.

Speaker 1: It's quite frustrating because I'm sure I've got some classic songs there but I don't actually write them down. Get up at 2 in the morning like-

Speaker 2: What instrument do you play?

Speaker 1: I play guitar. I sing, I'm a singer and [inaudible 00:17:13]. Good line for melody.

Speaker 2: It's quite an unusual mixture that the, kind of, economics and musician. Computer science.

Speaker 1: I'm not sure about the musician and computer science. There's a lot of structure, isn't there.

Speaker 2: No, I can see those.

Speaker 1: There's a systematic approach.

Speaker 2: But in way I can see lots of overlap of economic and computer science and I can see overlap- but I think it's unusual to find somebody whose got all three of them mixed together.

Speaker 1: I should be writing a computer science, I should be writing an app that allows you to play music and selling it. That's the ultimate combination.

Speaker 2: Or write an app that takes economic data.

Speaker 1: Yeah, and turns it to music.

Speaker 2: Makes music.

Speaker 1: There you go.

Speaker 2: New use for big data.

Speaker 1: Right. I've gone off topic now.

Speaker 2: And I'm thinking I should have. Go on.

Speaker 1: What do you think is the main purpose of the change from ICT to computing?

Speaker 2: I think it's that really. I think it's everything we just spoke about. Apart from the economics and music. It's that ability to get students that can solve problems. I think that's what it's trying to cultivate because well, it was, I don't know it's probably one of the questions that you're on. It's was kind of couched in the argument that we need people to protect us from cyber war. We're building soldiers with brains to be able to solve problems. I guess taking us back to Birchley Park and touring and people like that, that whole crew at Birchley Park who won wars by being or having that ability to think. I mean, when you explain to some of our students, not only did they decrease but they actually translated the language as well. Some of them, it's just kind of, well, that's what somebody done, I don't know what you're on about. But just to see that kind of, well that's quite an achievement.

Speaker 1: The fact that they had to understand a foreign language. They'd understand German well enough that to be able to decrypt something into German, then translate it out of German.

Speaker 2: It's a massive mind blowing-

Speaker 1: And do it or mostly at least, first by hand.

Speaker 2: Yeah. I think that's probably partly why they've got the idea but then it is an applicable skill to use. It's a transferrable skill, isn't it? Problem solving.

Speaker 1: Yeah.

Speaker 2: To look at the use a student might get in any kind of role or any job. It's a transferrable skill.

Speaker 1: What do you think about the ICT bit? About losing that?

Speaker 2: Losing it. No, I agree. See when I picked up this job 17 years ago. I was just trying to work it out there. We were switching from a more, probably a more- well actually it didn't really have a curriculum and I suppose it's probably just a bit vague and people probably teaching what they wanted. So when the national curriculum came in, it was targeted specifically at skill building the use in applications.

Speaker 1: Do you think it's moving away from skill building?

Speaker 2: I would say so. For me, there's an analogy where I don't know if I've expressed this to you before. That would be teaching children how to drive whereas the computer science is teaching them how to fix the car.

Speaker 1: Which is exactly the opposite what just happened in mechanics really.

Speaker 2: Is it?

Speaker 1: Well in a way.

Speaker 2: Right. Okay.

Speaker 1: Well if you think about how very few people fix their own cars now.

Speaker 2: Yes, they don't.

Speaker 1: And can fix their own cars.

Speaker 2: You're right. Yeah. Well we don't offer that skill anymore. It's not involved really.

Speaker 1: Somebody like me opens up the bonnet of my car and I say, "oh".

Speaker 2: There's a box there.

Speaker 1: There's a box there. A tube over there.

Speaker 2: Yeah.

Speaker 1: Oooh I can check my oil.

Speaker 2: Yeah, that's about it. Yeah, that's me. Oh I'll just clean that. It looks nicer clean, doesn't it?

Speaker 1: Exactly. I don't know what it does.

Speaker 2: Shiny.

Speaker 1: Exactly.

Speaker 2: No, I would agree. Yeah, fair enough. Yeah, we've become. All right, we're all drivers and we'll be increasingly drivers. I mean, how far that we're just driving these things, aren't we? We're not really fixing them.

Speaker 1: Yeah.

Speaker 2: If a monitor broke or the PC stopped working, you know, when they first came out and they said PC. I used that term very recently when I talked about BBC micros and Sinclair and Commodores and that kind of thing. Most people probably take that, somebody like Joe would probably take the box apart and try and work out what's going wrong with it. Or we'd be buying stuff and just say, "Ah, I could probably could fit it in". So, you know, we're even further removed from- And again, I think, there's a reason for this. If we want to stay ahead again, it is economic I suppose, I want to stay ahead of the technological game then we need to have people that can actually create the technology, not just use the technology.

But, yeah, going back to the IT computer science argument. We need drones. No, we do. You know people need those skills to do those other jobs. Say for example, that business comes up with that new bit of technology and it's been generated by somebody that does computer science. Who's going to do all the budgeting, whose going to be able to produce the reports? Whose going to have the skill set to generate that sales pitch? Yeah, I can come up with the words but I can't express them because I ain't got the skills to communicate them.

Speaker 1: Yeah.

Speaker 2: And I think that's going to be partly missed because I don't what- well actually, you're right, it's maybe going to come across in English because I know, perhaps more so English and picking up techniques. Maybe that's spreading into different subjects.

Speaker 1: I suppose if you think about it. Word processing, that obviously will transfer into English but if you look at something like using spreadsheets. That isn't being transferred into math, necessarily.

Speaker 2: No, it's probably more dangerous in asking this in geography and guess more people are supposed to be doing geography.

Speaker 1: Because there's still this resistance, I think, in mathematics to overusing tools.

Speaker 2: Yeah.

Speaker 1: That we should do it in a-

Speaker 2: Which is right because it helps this.

Speaker 1: Yeah.

Speaker 2: It is the methodology, isn't it?

Speaker 1: There is justification for it but someone like Mr. Wolfram, Conrad Wolfram, who wrote Wolfram Alpha, the search engine.

Speaker 2: Yeah. I don't most going that way, I was going to say, is that the same guy that did it.

Speaker 1: Yeah. He's British and he speaks about how we shouldn't, in mathematics, we teach too much calculation and not enough concepts. And he says, we should teach how to do calculation but we shouldn't require kids to do calculation.

Speaker 2: Well that's what computational thinking aspect is, I guess, isn't it. Are we teaching too much coding rather than how to solve a problem.

Speaker 1: Yeah.

Speaker 2: And I could argue that, that's probably what the new GCSE looks like. Why it's got that waiting towards computational thinking and only 20% of it's going to be course work but what's that 20% of course work going to look like. Is it going to look like impossible to achieve and does it really have to extend a value or should it really have more than 20% of value. If you understand what I'm saying.

Speaker 1: Yeah.

Speaker 2: Is it too much.

Speaker 1: How hard is it in comparison.

Speaker 2: Yeah, relatively in terms of it's weighting, is it too much.

Speaker 1: And is the computational thinking not explained enough that it feels like it's appropriately 80%.

Speaker 2: Yeah. I tell you what I will say. I mean P– and I, we haven't seen any enterprise less than 70%.

Speaker 1: But I've seen some of the stuff.

Speaker 2: The materials and stuff. Yeah. At the end of the day we spent- That was probably one of the hardest topics to deliver because it is so dry because it has that business element to it. Like we said last week. But in reality, it's prepared us perfectly well, I think, for communicating some of the concepts we're going to have to get across. And you're talking about teaching the concepts rather than the practicalities of it. And I think that's going to be big because we're in a computer room but the natural inclination is for students to want to go on a computer. It's not what we did with year 5 yesterday. They spent 20 minutes and they sat with a whiteboard and pend. They're all- You can see them twitching. You know, want to be on computer and they don't need it.

Interestingly, when the ICT curriculum came out there had been national curriculum in 19- When was it? When did I get here? It was 2000/2001. The QCA all started kicking off around and they put 15 units out there to spread across 7, 8 and 9. So much so that we still use some of the 9, 7 by 1, 7 by 2. But they put 15 units out there across Key Stage 7 to 9 and that was the worst suggestion in that you don't always need to be sat at the computer. But a lot of those things got watered down. There was always and it seemed to be a bit of a control element to it that got really wired up out of most people's curriculum. They weren't doing much in terms of systems and float jar, [flowall inaudible 00:25:57] and things like that, that you've already seen.

Speaker 1: Yeah.

Speaker 2: They weren't using that. It kind of- some people just like, "Oops, put that to one side and we just focus on this bit," which is perhaps maybe where the issue was because it was suppose to roll with the other aspects of it.

Speaker 1: The coding and the other aspects.

Speaker 2: Yeah. Well less. They probably wouldn't have gone with coding because it was very visual and it's probably very much what you could get away with at Key Stage 2, I don't know. I showed you the crocodile clips stuff, did I? No.

Speaker 1: I don't think so.

Speaker 2: I'll show you in a minute. I'll let you carry on. Sorry, because I will really go of-

Speaker 1: Failing to ask the questions. As I said, the questions are more a guide for me as much as I think. We've kind of covered it but do you think computing curriculum is relevant to other people you teach. Do you think that they should- The underlying idea of it being a curriculum subject in Key Stage 3 particularly, is that everybody should have these skills. Do you think that's right?

Speaker 2: Yeah. If we tap into that computational linking aspect of it, I suppose so. I don't necessarily believe that they will all want the other parts of it but whose going to fix stuff when it's broken? If they can't do it themselves.

Speaker 1: Yeah.

Speaker 2: Increasingly we end up, kind of, exponentially growing, we're kind of shrinking. It's got one person in whole of UK that can actually fix anything.

Speaker 1: Fix anything. Fix your house. And especially if we consider kind of [internte inaudible 00:27:33] things.

Speaker 2: Yeah. But the argument is that putting together methods of things, devices that don't need you to know how they connect. You just plug it in and it will work. And that's arguing what it does. So even so, if they won't need to understand the principles behind it.

Speaker 1: Yeah.

Speaker 2: They'll just need to plug the bloody thing in.

Speaker 1: Zaps in.

Speaker 2: Magically works.

Speaker 1: But isn't there an argument as well-

Speaker 2: Unless the broadband is crap.

Speaker 1: If you don't understand it, you don't necessarily have an appreciation of all that it's doing.

Speaker 2: Yeah, or value it I suppose. It's just expectation, isn't it then.

Speaker 1: Or understand, for example, that if your toaster is connected to the internet, somebody else is looking at how often you make toast.

Speaker 2: Yes, it may well be happening in the rest of your wireless network as well.

Speaker 1: Or happening in the rest of your-

Speaker 2: Own network, yeah.

Speaker 1: Selling your Kingsmill because you make toast every morning.

Speaker 2: Yeah. I mean there is that massive data protection application, isn't it really.

Speaker 1: Yeah.

Speaker 2: That what- Yeah, you right. I mean if you get an Itoaster, you can say.

Speaker 1: You can get a-

Speaker 2: Itoast.

Speaker 1: You can get a kettle that you can send a message. You can turn on your kettle from your phone.

Speaker 2: Well, I know you can. Yeah. Just like when you wake up, saying I'm having a little- tea's made and it will be on when I come down. Somebody's coming but there's no water in it, it wouldn't be boiled anyway.

Speaker 1: But it will be boiled anyways.

Speaker 2: I was going to say something else connected to that but it's gone. That's what I mean by improvising. Lost it but I-

Speaker 1: Interviews are more like Jazz.

Speaker 2: Yeah. Still trying to chat myself back there. No. I don't think so.

Speaker 1: I'm going to skip the next question because I think we answered it, which is how do you think people will use the skills and concepts they learnt through the computing curriculum in the future but I think-

Speaker 2: I think they'll apply it elsewhere. There was another good sound. I wish I could remember. Gone.

Speaker 1: In fact, we've answered a lot of these. This is why the improvisation works. What do you think is the most important thing you teach through the computing lessons?

Speaker 2: Well.

Speaker 1: In fact all the-

Speaker 2: I suppose the foundation of it is, is that computational thinking, isn't it really?

Speaker 1: Yeah.

Speaker 2: But-

Speaker 1: It's funny though because the idea of computational thinking is clearly important. It comes up in the literature a lot but it's not necessarily super well defined. And I'm not-

Speaker 2: This is where we try to define it in the HYP, in the first HYP session.

Speaker 1: And it's interesting that we still are having- I mean I've been to a number events over the courses of this year really and computing curriculum has been in place for two years. We've been talking about it for longer. There's a general feeling, that's the important bit but we still don't really know what it is and there isn't kind of-

Speaker 2: Well there's no computational thinking in hardware, is there? That the students being able to understand how that whole works by taking the box apart. There's no computational thinking needed and yet that aspect is important to some students.

Speaker 1: Yeah. Oh, I think it's important.

Speaker 2: More so perhaps than those ones that like that abstract way of thinking, which is what the rest of it is. Isn't it, really?

Speaker 1: There isn't, and it's more in delivery and because the curriculum itself is so general but there isn't necessarily a requirement to get hands on with a computer and pull it to bits.

Speaker 2: We do.

Speaker 1: Obviously it's possible to work it in but the curriculum says something like, know how it works.

Speaker 2: If we were to do a day based on hardware we would stay, which is what we do because we still teach them database principles. I suppose that probably one of the few things that's vaguely related. It's no longer necessarily in this, interestingly, although SQL is. I don't know why that isn't in the concept every day but it isn't. But it is something we still teach. We're going to teach students about hardware, we'll get them to create a database.

Speaker 1: Of the hardware.

Speaker 2: Of hardware, yeah.

Speaker 1: Yeah. But you're not going to start talking about what, focusing on the hardware itself. The hardware becomes a tool to teach.

Speaker 2: Yeah. We taking it by them, we taking apart a computer and they've had a good nosing around and found lots of stuff and done some research on it. Not really necessarily how it functions though because if we go at that kind of level then you'd be understanding how, which is what I was taught, was to how it writes a block to the drive and how it uses the addressing system. You know, just going back and say, that's what I like. I don't think that's something we've- well it's not in there anyway. I don't think. I've never checked. I think. You know, it's not something that I've felt necessary to teach students but it was what I was taught. It was explained to me how it writes a block and how that block gets an address and all that kind of thing. So I don't entirely know whether that's something that's going to happen.

Speaker 1: What do you think is the hardest thing to teach though in the compu-

Speaker 2: Aptitude. You know, we can teach it, can't we? It's the same as, I think for some penny drops, others, penny takes too long to drop. In the time that we have. It may drop eventually but I don't necessarily believe.

Speaker 1: What do you fell like you have to spend the most time teaching?

Speaker 2: We should spend more time teaching computational thinking. We should be and that's something that we aren't doing yet. We should be enforcing it in the project documents that we create and I don't believe we're doing that yet because that's very much for thinking of five years. Well I've always thought of things as five years because the skills we've taught at Key Stage 3 have always informed the GCSE. By the time they've got there they're used to it. It's routine and that arguably explains the resources we get for GCSE IT. We're going to need to do more of that for computer science. We've talked about computational, we talked about that in year 7. So we're doing that at the beginning of year 7. Now what we doing with year 8 because we missed it with them in year 7. We'll do that at the beginning and then project should have, what is the problem?

Speaker 1: Yeah.

Speaker 2: You know, write me an algorithm, write me Sudo Code. That's the format it should take and if it does then that generates the foundation.

Speaker 1: What do you think if you were, sort of, to identify one thing that you spend the most time on now. What do you think it is?

Speaker 2: Coding. I'm not 100% sure that, that's necessarily right but-

Speaker 1: There is the argument that Miles Barry, who does lots of stuff, who's a professor who comes up when you start looking at this stuff, says he promotes using coding because he says-

Speaker 2: We're still going to practise it.

Speaker 1: His argument that he comes back to all the time, I've heard him make it a few times is, computational thinking is the important bit but the best way to teach computational thinking is through coding.

Speaker 2: Is via coding, yeah.

Speaker 1: And so, we should teach coding. And then he goes on to say, we should teach coding in every lesson, and has great examples of how to use in Martin, math but that's the argument for teaching coding.

Speaker 2: Yeah, I suppose.

Speaker 1: But it's one, it's an argument. It's not the only one that's out there.

Speaker 2: I just pick of myself. Never happy.

Speaker 1: It's coz you're an artist really.

Speaker 2: I think. Well it's all, it just evolves.

Speaker 1: Okay. What would you change about the computing curriculum? If anything?

Speaker 2: Well, not the curriculum. It's the delivery time, I think. It'll never get there. It never has. ICT's never been given the opportunity to get more than an hour a week. If we had more time what wonders we could do.

Speaker 1: Do you think that the pupils need it? Long term, short term.

Speaker 2: If we took everything that we teach them and say, but look, this is a fun way we can do and then we can do it this way too. And we could give them, all of the, I suppose a taste of something that interests them but we can't do that when we've pressures that require us to work towards-

Speaker 1: But do you think they would benefit from having more than an hour a week of computing?

Speaker 2: Yeah, I would say so because we talk about all this. I mean, last night for example, our coding club, we set the wireless network with the micro bits. If we had more time we could have done that this year with year 7, after we done the network unit. This micro bits, oh, lets come back to this. Micro bits are here, they only got- we actually got them sent in packets. Well I don't know how big they were because radio, isn't it. But we got them sending technically packets and we had great fun last night. It took us the hour to do but by the end of we just used new Microsoft environment for micro bits and-

Speaker 1: Yeah.

Speaker 2: One of the students worked it out and I said, "I can't remember how Joe did this. Hang on." And I sat there because I went to the cession with the conference. And one went, "I've got it. I'm got it working." Cool. So, one of my year 7's worked it out. That's what I would like to have been able to do.

Speaker 1: And that's the type of things where you can. Every child probably has a wireless network at home.

Speaker 2: Well, but the thing that I was saying, that with extra time it's, here's some stuff. This is what we're trying to solve. Somebody come up with a solution. How can we do that.

Speaker 1: That's something that I think-

Speaker 2: Lets play.

Speaker 1: Is quite interesting about, I mean, the way you use success criteria and I was watching P– and it's a really powerful way of teaching. The, here's your problem, break it down into the steps, break it down into the success criteria.

Speaker 2: Yes, and they kind of flow from each other.

Speaker 1: Yeah, and I can see the power of that and, I think you came in, one of the things we don't teach overtly is project management but it's so important.

Speaker 2: Sorry.

Speaker 1: And on the other hand what you noticed about that is that there has to be, because it's within the structure, there has to be one solution because it's in a school setting where you kind of need a single solution that can be marked.

Speaker 2: You just got me thinking that because, I mean, technically success rates are really harder than algorithm.

Speaker 1: Yeah.

Speaker 2: If you look at that point, I need to write the algorithm to solve that.

Speaker 1: What you were saying about using computational thinking, what you already do, you just have to repackage it.

Speaker 2: Yeah.

Speaker 1: And repackage it as computational.

Speaker 2: Or decomposition when I said, algorithm, right. Is the description. Decompose that. What will successful switcher look like. Now we've decomposed that into a successful solution.

Speaker 1: What's the algorithm.

Speaker 2: What's the algorithm.

Speaker 1: What's the algorite of them you will follow as a person.

Speaker 2: But it might be three, it might a separate one, broken up for each individual problem within that. It needs to be able to do that. How do I do that? Is there algorithm for that and that then just click them together, which is what I was ranting about this morning, was at how annoyed I was that none of my computer science group were even following the decomposed problems. They weren't even- anyway. It's another runt.

Speaker 1: But these are kids who haven't had.

Speaker 2: They haven't had this. But I did decompose it with them.

Speaker 1: With them. Just beat them over the head with it.

Speaker 2: Well what I mean is that they weren't doing what the task was asking them to do even though it was there in front of them.

Speaker 1: Yeah.

Speaker 2: "Oh, I haven't done that." "Why the hell not." "I didn't know I had to." "Yes, you did. It's there." And then somebody else was like, because you just sat there like that for an hour. The boy was in with P–, which is why I didn't runt. Was in that class. There are more people I want to runt about but he just sits. Clearly it might be something else happening there a little bit, it looks a bit mean but he just sits there for an hour like this. And looking at other people's work and that's it. I'm thinking, what the hell?

Speaker 1: And he's doing a GCSE in computing.

Speaker 2: What the hell? Anyway, moving on. We going to have to move on.

Speaker 1: Yeah. I'm noticing.

Speaker 2: From thinking that. Just from thinking that.

Speaker 1: I'm going to skip the next two questions because I think we've answered them but I'll read them so that you have the chance to make that judgement .

Speaker 2: Yeah.

Speaker 1: If you can only teach one aspect of the computing curriculum, what would it be?

Speaker 2: I suppose if you make it entertaining enough it would be just pretty much everything I'm doing now with GCSE, which is problem, algorithm, Sudo Code, test, trace table, problem algorithm, Sudo Code, trace table and just that, and then just give them lots of problems to solve.

Speaker 1: And you would step back from the code? From doing-

Speaker 2: I don't know because if Sudo Code it didn't initially matter. It is arguably translatable into any code then.

Speaker 1: Yeah. Sudo Code's funny because it's again, where there's terms that is [inaudible 00:39:59] quite a bit but is defined differently by different people.

Speaker 2: Yeah. I had to teach them yesterday. This is our exam board. What I've decided to go with this exam board, I have. Even though I've got books for all them. I'm going to go with these guys.

Speaker 1: So the exam boards essentially define what Sudo Code.

Speaker 2: They have decided to find their Sudo Code each.

Speaker 1: Each has done it. Well, because, I mean, we ran into this as a code club because we had a child who came to our Saturday code club and said, "All right. There's an assignment from my teacher. I have to write this thing in Sudo Code." And so, I said, "Fine, we'll help you." And one of colleagues-

Speaker 2: Came back and it was wrong.

Speaker 1: Used to teach at that school and he came back and it was all wrong.

Speaker 2: Yeah.

Speaker 1: And you're like, "Well, what we gave you is Sudo Code."

Speaker 2: Yeah. It's a version of. And I said that to my lot. I said at least taking away and turn around and say, you know, "You don't have to stick to this Richard thing."

Speaker 1: But this is, yeah.

Speaker 2: But this what we're suggesting. And this is what teach, this is what we want you to do. And I think the questions in the exam are going to be written in their Sudo Code. So.

Speaker 1: Well at least you know what it is then.

Speaker 2: So I've had to teach them, yes to Sudo Code to translate our flowcharts. This is where we are. Flowchart, translated it into Sudo Code. But given-

Speaker 1: Again it's one of these big concepts that seems key and yet isn't being pinned down to be universal.

Speaker 2: Oh hell, no. Lots of people have kicked of. Lots of teachers have kicked off without even playing it. You know, you can't define a Sudo Code. You can't define a method of writing Sudo Code. Well we are doing. Just jump through that one. But what I'm going to do when I do teach the coding is, that is very much that process that I just described. Even for a simple, how do I retrieve, how do I take input from a user and store it. A simple input command in python. Variable equals input, straight in, whatever. Simple in book command from python. I'm going to expect them to do a flowchart for it, Sudo Code it and then python it. As laboriously as that and it will seem a little bit mhhuu. It just comes to the drill and that more like it.

Speaker 1: And those first three steps will be consisting whatever language they're working.

Speaker 2: Yeah, they should be.

Speaker 1: The flowchart.

Speaker 2: The methodology.

Speaker 1: The Sudo Code flowchart and then coding it. And it shouldn't actually matter. I mean, you could even, theoretically allow them to choose, which language they wanted to work in.

Speaker 2: Yeah.

Speaker 1: And say, "Okay, you like python, you like C, you like." Find whatever.

Speaker 2: Yeah.

Speaker 1: As long as it works and as long as you've done those first steps.

Speaker 2: Yeah. I got you, right. But I'm assuming that's what the NEA's going to expect them to do.

Speaker 1: NEA? National-

Speaker 2: Non Examined Assessment. Cross word. So that's what I'm trying to prep them for, more than anything. The teams increasing like, it's going like an MFL because they've got no access to any resources.

Speaker 1: MFL?

Speaker 2: More of our languages. Like a speaking test.

Speaker 1: Right.

Speaker 2: They've got no access to any resources. They can't take any materials in with them. They can't access the internet. This is all so unrealistic as well as the Sudo Code being unique. And they have to sit in silence and work and that's it. That is-

Speaker 1: That's how they'll be tested on this.

Speaker 2: That's how they'll be expected to be doing this.

Speaker 1: And that's the only thing their GCSE create is based, is that?

Speaker 2: No, that's was two papers. One computational thinking and the-

Speaker 1: And the other 20% is just that NEA. It's called an NEA. Non Examined Assessment.

Speaker 2: Right. Got it.

Speaker 1: And that will be done under exam conditions. It's unrealistically for a coder. They will not be able to look up anything, use any references, any previous examples that can be followed.

Speaker 2: You can bring their own stuff.

Speaker 1: Nothing.

Speaker 2: Yeah. It's completely different from that.

Speaker 1: So, if you were doing model four languages exam, you'd have to remember your speech and say it.

Speaker 2: But you think that even in language, there is a certain amount of logic to that. If someone comes and approaches you and asks you a question. If you're in a foreign land, you know, I've been, I've been in Germany and-

Speaker 1: But you can pull out a phrase book.

Speaker 2: But you can argue that, oh what if you're caught without it. What if you forgot your dictionary at home. But in coding it would be very unlikely that you would be sat coding something.

Speaker 1: Without access to the internet.

Speaker 2: Yeah, exactly.

Speaker 1: I'm sure Russia or China or some kind of computing gulag.

Speaker 2: What is the point of doing it. Why are you coding something. This is not the best solution for this.

Speaker 1: That's it. That's how it's going to work. So I have to prepare some- Part of that process that I'm preparing from the wrote process, which is pretty much what it is, is even at that very simple level. Lesson 1, we're going to go back to looking input in python. "I can already do that, sir." "I'm sure you can." There's your problem. Flowchart, Sudo Code, solution.

Speaker 2: So it's teaching them a process rather because they're going over things that they already know.

Speaker 1: And then when we get higher up. It's a more challenging one, decompose it.

Speaker 2: That will be at year 10 or 11?

Speaker 1: Yeah. Year 10. I'll be doing that in September.

Speaker 2: At year 10, refreshing what they've done in year 9 but-

Speaker 1: Building on that and then we're going to start looking at arrays and file handling and obviously I think that the key for most of these Non Examined Assessments is going to be multidimensional arrays. So they're going to need to understand that, you know, reference something in 2D rather than in a straightforward list.

Speaker 2: Yeah.

Speaker 1: That's what kind of worries me a little bit.

Speaker 2: That one that, the thing abour arrays, if you got one dimensional array lists, you've got two dimensional arrays, you have three dimensional lists and then you can actually have more dimension as well because it's not in real space.

Speaker 1: Yeah. I mean that what's worrying me a little bit. That's why M– should go back to teaching matrices because then they'd be okay with that.

Speaker 2: Yeah.

Speaker 1: Which is thankfully why I am, that I did, sort of that I'm a [consternated inaudible 00:45:52] techniques, and my economics degree, concepts, and matrices. I'm glad I did that because you know that conceptually it kind of fits with what I'm going to need to do now. When you mentioned to me about who you'd come across the Matrix. I don't know what you mean.

Speaker 2: They're probably even too young to know the film.

Speaker 1: You've got a two dimensional array in front of coordinates.

Speaker 2: Yeah.

Speaker 1: Yeah. That's "Oh, the Matrix." Anyway I diverge, I go off again.

Speaker 2: It's fine. How do you think, well regardless actually. So I'm going to skip the next one because I was going to say, how do you think that you're judged and evaluated on your delivery of computing but I feel like actually-

Speaker 1: It's results orientated, probably, ultimately. But enjoyment, engagement. I will give it our due, that's what we have to try and make it be like. Progress, evidence of progress, which is increasingly tricking when it's completely new. How do you prove.

Speaker 2: Do you think that's an aspect of it being new or an aspect of it being computing.

Speaker 1: No, I think for me, at the moment, it's not expected to be new because I can't, well I can.

Speaker 2: I mean, it's the challenge of the new curriculum.

Speaker 1: I can't visualise how that would happen but right now I don't want to because it sounds like too much work. It can be passed on, quite a lot of work on that.

Speaker 2: Yeah, he's working on the assessment criteria.

Speaker 1: Thankfully that is well in [inaudible 00:47:23] but it's a lot of work. It just brings me back to say 15, 16 years ago when the new curriculum came out and I spent many years battling to get to something that fairly optimal and now it's gone. It was a 15 years wasted. I don't know. Well probably not because I've sent students out there with aptitudes and abilities that they probably wouldn't have already had.

Speaker 2: And that they're using.

Speaker 1: That they are using. It still makes you feel like [inaudible 00:47:50]

Speaker 2: Yeah. Because-

Speaker 1: It's such a radical wholesale change.

Speaker 2: Yeah.

Speaker 1: I mean you can say jump if your English, History. I mean these are things that haven't kind of changed that much.

Speaker 2: And I suppose at this level it's more of a change because if you look at, kind of, Key Stage 1 and Key Stage 2 computing, there's still an awful lot of the ICT stuff mixed in there.

Speaker 1: And there will be for a while when they try and slowly phase it out because they don't want to do the wholesale.

Speaker 2: They do the numeracy in the-

Speaker 1: Oh we suppose to do that too?

Speaker 2: Yeah.

Speaker 1: Everybody else can do that because they're all using tablets. They can do that.

Speaker 2: It's a funny that because there are in primary school. You can kind of talk about doing things cross curricular especially-

Speaker 1: Or natural.

Speaker 2: And that seems to me, from what I see, much, much harder to do when you come get to the second phase.

Speaker 1: We come from ICT across the curricular pushes into doing, everybody should be using spreadsheets by processing a power point across. Now it's like, everybody should be understanding how to safely find information and use services like. Why not just tell them that's what they're suppose to. Why not put it in their curriculum that they need to be teaching that. Why tell us that we should be delivering that. If you know what I mean.

Speaker 2: Yeah.

Speaker 1: Because we would do that anyway.

Speaker 2: Well, yeah. I mean, for example, why should you teach, which translation service is best when it should be the MFL people who say, "These ones do a really good job."

Speaker 1: Or none them, usually.

Speaker 2: If it uses an algorithm, no.

Speaker 1: That's one of the first ones, yesterday, I put it up from BBC on their website. I think it was on the tech section that I put translators and things and it says, the translator, "Please do not eat the carpet." I'm sure they didn't mean to do that but that's what it actually said on the sign, "Do not eat the carpet," or "Please do not eat the carpet."

Speaker 2: But again, it's something that the MFL people can have that conversation because they could do a good translation whereas it will be very difficult or more difficult for you to do it.

Speaker 1: All that I've got to do is put a survey out and go, poke, poke, poke. Have you thought about this? And you're teaching. Well you shouldn't be doing it. You doing it? No, you not doing. Well you are.

Speaker 2: That always goes down well with teachers the, you should be doing this thing that you didn't think about. What do you think are the priorities in terms of, kind of, your Andy and off stead, when they're looking at what computing should be delivering, should be doing.

Speaker 1: I don't think A– would necessarily tell us what to do. I think he trusts us on that to do what we're expected, we need to do but as far as his support, without a question is extremely-

Speaker 2: But why do you think. Does he- You've said before that-

Speaker 1: You've seen. He's sorts himself out with his children. I think primarily but it is a science. He understands its important. You know, he's a clever man.

Speaker 2: But he wouldn't necessarily tell you. He trusts you guys as-

Speaker 1: To teach what I have to teach. Yeah. Pretty much. I mean it's on the micro bit thing that we did on that evening, on that Monday and he was just like, wow, how you going to follow that next year? I don't know that I am. I thought that was it. I was going to do it only once really, I wasn't planning on doing anything else. Yeah. It's just a one off. I was inspired and that'll do. It was a shit load of work. So I'm not doing it again.

Speaker 2: No thank you BBC.

Speaker 1: So, that, you know. So he sees that result.

Speaker 2: But do you feel like, and I'm not super aware, I don't know hugely about how [inaudible 00:51:31] does secondary schools but do you know what they're looking for? Do you know what they expect from computing?

Speaker 1: All that probably. I don't necessarily know whether that's what they'll be looking for but they'll ultimately be the same as, I would assume, because we wouldn't necessarily get a specialist watching us. They'll just be looking for the key indicators. You know, engagement, differentiation, progress.

Speaker 2: What do you mean by differentiation?

Speaker 1: Students being provided with different work for different abilities, which is partly why sat with them anyways so we can minimise the amount we had to put into that approach. But when you've got mixed ability it's quite tricky to do that. So yeah, that's kind of what they're looking for. I would say. Well they still looking for the same old markers. Saying the same old markers, that's a bit mean. But yeah, that kind of range of different things, active learning, independence, understanding where they're going, the targets, where they need to go to get to the next point to progress. All of those things I would say but coached within this environment.

Now they won't all know what this looks like in an outstanding lesson. They will know what an outstanding looks like in a vacuum but not necessary for a specific subject buts that's how they roll. That's how they send people out and they send people out that are specialists but all specialists in every subject.

Speaker 2: Yeah.

Speaker 1: Not anymore. They used to send someone for everybody, in the old days it used to be one each.

Speaker 2: One in each area.

Speaker 1: We're going to get kids coming in here. What do you want to do?

Speaker 2: That's fine. We can stop or we can go somewhere else or what?

Speaker 1: Well, I'm free after lunch if you want to pick up or stay.

Speaker 2: Yeah I'm just going to pop back to-

….

Speaker 1: To what extent do you think that the computing curriculum relates to how the pupils use computers in their everyday lives, or how do you think they do use computers in their everyday lives?

Speaker 2: Interesting.

Speaker 1: And ask them about this one.

Speaker 2: Yeah. The only people I can think that would benefit from that would be those that play RPGs, I would say, or strategy games. That's it, in that gaming context. Because, hmm. If it's a strategy game I guess you can start to think about what is it I need to try and achieve here, what's my best route for achieving that? Especially if it's about moving and positioning players, so to speak. With RPGs I guess it's just a little bit more that guy keeps shooting me, how am I going to stop it happening next time?

I don't know, other than that, do you know I'm trying to think of another application that they may get from it. Because I think most, and it's probably a little bit disingenuous of me to say this. I think a large number probably use them pretty blindly. Like most adults do. They use them fairly blindly and they expect them to work and when they don't they cry. When in reality they should just be using them with open eyes and then they do work and then they're happy, they've got [inaudible 00:01:19]. It's not about it working, it's about it not working is probably-

Speaker 1: Do you feel like there's a pretty low, like I would say ...

Speaker 2: It might be really disingenuous of the way it impacts on students.

Speaker 1: Do you think they use computers a lot? What would you say they use computers for?

Speaker 2: I would argue that they use them increasingly less because they come in here with less PC skills, shall we say, personal computer skills. The kind of thing you'd need to use to navigate the simplest of personal computers. Whereas, actually again that's probably the wrong definition, isn't it? What I mean by that is what we would've traditionally called a desktop and a laptop. Whereas most of them are quite happy using a personal device, shall we say, which we could define as a phone or a tablet, so that they come in with a massive gap in skills.

Speaker 1: Do you think that matters?

Speaker 2: Yeah, because then they're not ready to use the tools that we need them to be able to use.

Speaker 1: Do you think that will always matter?

Speaker 2: We could argue, let's just switch to tablets and they can do all the coding on tablets, but I don't necessarily believe that that's ... It kind of defeats the objective, the purpose ... Well, what we were arguing before or what we were discussing before about the purpose of the curriculum, it would defeat the object if they were using a car. Driving a car to learn how to fix the car. It doesn't really seem to make much sense, then, does it? This is about file folder structures and they're missing all of that. That's part of how it functions, isn't it? It's part of how to find things.

Speaker 1: So you think in some ways, so much of the functionality of contemporary devices is hidden in a way that it wasn't with ...

Speaker 2: We're further abstracted, to use that kind of phrase from understanding the architecture or the functionality of the structure.

Speaker 1: If we go back and think about Windows 95, I guess, if you go back to 2000 in Australia, or even-

Speaker 2: Yeah, we were a bit closer. I think we were running 98. We were using Novell when I got here. So it's a completely different kettle of fish, yeah.

Speaker 1: I have never even heard of Novell.

Speaker 2: It's a network operating system, so it wasn't really a particular Windows base, it was just a network operating system and it just ran on clients, pretty much.

Speaker 1: Oh wow.

Speaker 2: We migrated from that to Windows pretty sharpish I think, because back in at the same time as the network manager. [inaudible 00:04:11]

Speaker 1: But the ...

Speaker 2: Doesn't exist. It did word processing.

Speaker 1: But those other forms hid ...

Speaker 2: So increasingly as you move away from shall we call it MS-DOS to where we are now, yeah they increasingly hid that magic that happens underneath it. But I would say even more so now. Because you know, there's not even the need to interact with it in any other way than just going like that. Which a pigeon could do, really. A pigeon could never move a mouse or tap on a keyboard. But a pigeon could definitely open an app. Well, actually, apart from the requirement to have some form of electrical conductivity between the app and the screen, or the screen and the digit.

Speaker 1: I don't even want to get into whether or not a beak can work and be conductive. Capacitive touch.

Speaker 2: Yeah, sorry, capacitive touch, that's the word.

Speaker 1: Does a beak have enough conductivity to ...

Speaker 2: It's an interesting thought, that, though, isn't it? Can I get a ... All right-

Speaker 1: Let's start with parrots.

Speaker 2: But you know you can ... It's one of my favourite [inaudible 00:05:24] because you do see the students, when something's not working.

Speaker 3: Yeah, this is where I'm like-

Speaker 2: Used to that. Where were we? Yeah, so [inaudible 00:05:34] because people just ... It's that idea that if I've clicked on it, it doesn't work the first time and they just peck at it like a bloody pigeon. Like a pigeon pecking a fag butt. This isn't food, this isn't food, this isn't food.

Speaker 1: So what you're doing in your computing lessons might seem so distant?

Speaker 2: I guess, in some ways it's very alien to some of them. So I don't even know what degree they've used any form of input device other or both, I suppose, input/output device is in touchscreen. I don't know whether they've used anything other than that. So they struggle with the mouse some of them. They struggle with the keyboard, the typing skills aren't there because they don't need the typing skills. Even though they'd be all right like that, but that's not-

Speaker 1: But it's interesting because there's loads of things that it connects to because first of all I've heard the same thing from reception teachers who have said that they used to be able to expect that when a child started reception they'd used a mouse and a keyboard, and that has gone away. That now five year olds all know how to use a tablet.

Speaker 2: Yeah, well, it's the easiest thing to do. Teach them how to push a button or just give them something to push until it works.

Speaker 1: Yeah. Again, a three year old can use a tablet.

Speaker 2: When I used to sell the toys that they're talking about that would prepare them for that, just before I sold the other stuff, when I was at [inaudible 00:06:55]

Speaker 1: This mystery toy. I have no idea what toy it is.

Speaker 2: I'll find it, [Felix 00:06:59]. It was geometric, pretty much shapes and stuff. But I am ... So I used to sell V-tech products. So obviously they've kind of gone back that way a little bit. They do do touch tablet replicas, but they still do the laptop type stuff. And he used to sell that stuff. Going back 20 years, 22, 23 years, that stuff is markedly different now than it ever was. Obviously technology has come along quite a lot then, but it's still ... There's not much button pressing. You're pressing quite a lot of tablet type interfaces.

Speaker 1: Well, I think that those sorts of ... Even in the last five years, that's changed. Where there used to be a market for these kind of pseudo-computers for very young children. And now you just don't ... Because tablets are cheap enough and available enough.

Speaker 2: Oh, I've got this one, I've upgraded, what am I going to do with it?

Speaker 1: And you can get then-

Speaker 2: Life box around it.

Speaker 1: Exactly. And you hand it to whatever age. They can-

Speaker 2: Because they're never going to want to take that apart and have a nosy around and see what's inside. Well if they do, they've broken it already. But they're never going to want to feel the need to do that.

Speaker 1: Yeah. You're going to have to start taking iPads apart. There's almost nothing in them, by the way. I've seen inside them. There's just nothing. Incredibly little in them.

Speaker 2: Yeah, because when we discuss hardware, we talked about hardware before, we talked about addressing drives and stuff like that. When we take the hard drive out and all those other bits out, and they just go bump, big lump on the table. They go, same as that tablet. Micro:bit, Raspberry Pi, these all have all these components in it. They're like, "Huh?"

Speaker 1: That's the other thing I was thinking about, is that with the Raspberry Pi one of the arguments I have with Kevin actually is he doesn't like the Raspberry Pi because it requires too many other bits. But we talk about whether or not people think of a computer as a screen and a keyboard, or they think of a computer that does computation. And actually from what you're saying, there's a shift even further where a computer is becoming a screen. It's not a computer unless it's got a screen. Then these things that have screen and keyboards or ... Those are all kind of things that they would use.

Speaker 2: Most of them would not therefore recognise a fridge or washing machine as those kind of things, although increasingly fridges and washing machines have more complex screens and keyboard. So eventually they might start to make the connexion. All right, that's actually [inaudible 00:09:34]. The swines, instead of programming it all in for me, I'm having to do it myself. That's an interesting thought, isn't it?

Speaker 1: How do you think that learning computing will affect the pupils' decisions about their choices about the future?

Speaker 2: I think some of them will definitely not be doing computing or computer science.

Speaker 1: That wasn't really the point.

Speaker 2: I don't know. I don't know really. I think at the same time, the same way that we always caught those that were interested in IT, because there's still to a great degree right now, students still see us as IT too. Whatever name you want to put to it. They still see us as something they're either going to love or loathe. They get the experience, hopefully, that they're going to love when they have us. And by that I mean when they have myself and P–, but I think ultimately we were going to get them all or we're not going to get them, whatever the case. I don't think it's going to much change about what they opt for in regards to us. Although it does increasingly become harder when the students realise what they've got to do is harder than just driving the car.

Speaker 1: So they come in and they expect it to be something and it's harder than they expect?

Speaker 2: I think increasingly it's getting harder for them.

Speaker 1: Do you think it changes how they ... This is kind of what we were talking about a minute ago. How they relate to computers in their lives, and this is broader than just the computer sciency stuff, but all of it. Do you think they take it onboard?

Speaker 2: I think some of them, they'll think differently, but again it would be the same as those students that get math. There's others that go away and go, "Yeah, 2 + 2, I can do that bit, that's fine." But I don't think it would change-

Speaker 1: But do you think they relate to ... Again, thinking more broadly. Do you think they relate to other people online differently? Do they use the internet differently?

Speaker 2: I don't know whether they sit down and think, "Actually, this isn't the internet. This is the world wide web. I'm just accessing the internet through this medium, this browser." I don't know whether they'd think down and think about that. I like to think they did. I still have an anecdote available where we talked about networks and our network stopped working and they were able to unpick what was wrong with the network, and that was our year eight as they stand now, so it would have been year seven at the time. So they were able to work out what might have gone wrong. So perhaps that might influence the way that they work at home, because at the end of the day you get your parents and they still get my, my poor brother, because now they've switched to Mac.

It was Windows before. They switched to Mac now so it's my poor brother. It's just kind of, how does this work? I can't get this to work. This won't happen. And stuff. Anyway, it's always been one or the other with us. Yeah, so, luckily it's my brother now. So I guess they're going to be the same. They're going to be the same. They're going to be those people. Maybe that whole generation, that have just used tablets and devices. With any luck we may have caught the tail end of the [inaudible 00:12:45], maybe it's the middle of them, who knows where we are with that kind of timeline.

But it might mean that those students will just about be there for those people at the other end who really can't fix that or really can't work it out. Like I was saying before, in the end it was just like one person that can fix anything. But these people, they made ... We may just have caught them.

Speaker 1: So they had just about enough understanding.

Speaker 2: Yeah. But they're not ... Now, you know, whereas if we just left them as they were perhaps not, but they're not designed to be fixed, that's the problem.

Speaker 1: The computers?

Speaker 2: Not anymore.

Speaker 1: No. Neither software or hardware is really-

Speaker 2: It's just disposable. It's not even [inaudible 00:13:22], no longevity in software at all.

Speaker 1: No. And well there's ... The hardware is developed but the software doesn't develop as quickly as the hardware.

Speaker 2: No, that's the irony really, because some of the hardware out there would make things completely different again.

Speaker 1: Yeah.

Speaker 2: Yeah, I don't know. I just have this prophecy of doom now. That you've made me think of. Too much science fiction, you see I keep going back that.

Speaker 1: The singularities.

Speaker 2: Yeah, so anyway. I don't know. I find that question quite difficult to answer.

Speaker 1: How it's going to affect their choices.

Speaker 2: Because as far as they're concerned, the device that they will probably end up using is just a commute, isn't it? It's just that, going back to the car analogy now. So whatever they're doing at university is just a tool, but it doesn't really matter how the tool works. It's just a commute. It's just going to get me from A to B, but I don't really care how it works. I'll just buy another one if it breaks.

Speaker 1: So it doesn't matter how it works?

Speaker 2: I would imagine that there'd be a lot of students that even having gone beyond this that that will be the case. Maybe if we're lucky enough with more time, we said that already, that to explore and experiment, that might switch a few more of them on to it.

Speaker 1: But you think ...

Speaker 2: At the moment.

Speaker 1: What we teach in computing, but again it means that what we come back to is that it's really about that conceptual computational thinking.

Speaker 2: Yeah. If they can take that [inaudible 00:14:57]

Speaker 1: If there's anything that they take away from it, it will not be the code, it will not be how things work, it will be a methodology for solving problems.

Speaker 2: Yeah. Which means that they could therefore apply it to anything in many roles over many different things.

Speaker 1: And that they could apply it to any software, hardware, or whatever.

Speaker 2: They could apply it to laying bricks, I suppose.

Speaker 1: In theory.

Speaker 2: If they can see a pattern.

Speaker 1: So that comes to the nice last question actually, which is thinking forward. In five years time.

Speaker 2: Yeah, that's what I was going to go to after you asked me the question. That one. You know you were saying where it would be in five ... If you came back to me, would it be any different?

Speaker 1: But also ... Well, two questions then. One is, in five years time is it going to be any different? But also what are they going to remember about learning about computing in key stage one, key stage three, four? What do you think is going to have the most impact, if anything? I should never ask teachers about the future.

Speaker 2: No, no, no, it's fine. Well, five years, I would say no. It's going to be about what's happening lower down, then, isn't it? Really that's going to influence what we're able to do now. But at the same time we're being pushed back from the [inaudible 00:16:07]. And hopefully, and I don't really know, you probably know more about what's happening at the primary level probably than I do really. And hopefully we're getting that which is going to scarily influence what ... So imagine this is us in key stage factors, let's use some props.

This is us in key stage three and four, and then we have this key stage two coming up here. The theory, the practise and the skills should mean that that means we're going more in that direction towards key stage five, but in reality key stage five is going to be in this direction anyway. So at some stage, the difference could be that the whole subject means that they are getting much a different diet out of it. So they're prepared for things that we don't need to teach them anymore so we can do more with them. In different kinds of ways.

Speaker 1: So do you mean conceptually or skill wise?

Speaker 2: Probably both because that's what we always ... Historically that's what has happened, really, with ICT. What was happening now was it was weak there, it was happening up here and it just went like that. And there's a point, what we've just left was probably an optimal position with ICT.

Speaker 1: It was just working.

Speaker 2: Pretty much, because we were able to teach concepts because they were able to pick them up further down and we were able to really drive things forwards, you know.

Speaker 1: This is kind of a limited perspective, but thinking about [the casa 00:17:29] then this conversation and the work I've done, really what we need to do is ensure that pupils in primary school are taught computational thinking skills.

Speaker 2: Yeah. We could bypass that.

Speaker 1: That everything else, it doesn't matter if they use Scratch, but they can. It doesn't matter what sort of algorithm, but they need to learn those computational, and it really is in my mind at least, about having teachers willing to pull away from teaching computing with computers and embed those computational thinking skills.

Speaker 2: Yeah. Brings me back to that student [inaudible 00:18:13] and what we could have done, I want to stop this thinking, I want to start to do some coding. But then all right, I'm using computers to teach computational thinking in my GCSE class. It's a shame we haven't seen some of the stuff I've been doing. You'd probably just liked us to have dropped in just to see where key stage three has gone to. It might have been of use to you.

Speaker 1: What do you think the kids will remember in five years of learning about computing, if anything?

Speaker 2: If they've opted for it, I would imagine they've already started on a path that would mean they're not going to forget it. But if they've not opted for it ... Don't know. Maybe they'll just remember how enjoyable the lessons were. Maybe that will help.

Speaker 1: Of course, that will help.

Speaker 2: About applying those skills? Again, maybe. But then you could say that about so many subjects that they're not interested in. They could move away from ... If they've opted for History instead of Geography there's obviously a reason why they've done that. Will they remember how to read a map? Probably.

Speaker 1: Hopefully. I hope everyone can read a map.

Speaker 2: So will they remember how to do some basic computing kind of skills? Probably. What would that be? Don't know. Would it be, maybe they'd subconsciously use those problem solving skills that they've learnt.

Speaker 1: I think it's probably the power of the [inaudible 00:19:36] those computational thinking skills into things like the top sheets. Because then there's that recurrence of over and over again looking at using it again.

Speaker 2: I mentioned that to P– before and we both agreed that it's probably the way forward. So just rewriting everything again.

Speaker 1: There you go. You're welcome.

Speaker 2: I think that can wait till next year. The deeper, the longer, the more, the further we go down this route of bringing computer science back, the more I remember from what I was taught. That was 30 years ago. So I guess if it's needed, it will probably come to the surface. If it's not, and they've not really pursued it then perhaps it will just ... It might end up subconsciously implanted in here as a skill that they didn't realise they were using and that's where it came from.

Speaker 1: But do you think if I asked them that, they would say ... If I said, "What do you remember?" Time warped five years in the future.

Speaker 2: What do you think you'll use or remember? Yeah I think a lot of them are probably saying ... Depends on which group you choose as well I think. Because you've got to bear in mind even something like an underwater, I've had a student in my [inaudible 00:20:53] that have just gone, he's going to do biology, underwater biology, whatever they call that. It's got a proper name, hasn't it really?

Speaker 1: Marine biology.

Speaker 2: That will be the one. And he's going to go and do that but he knows he's going to have to maybe programme robots and submarines and stuff like that.

Speaker 1: Where did year nines of that? I mean that is part of the ... Again, that comes up in the computational thinking literature, but also the whole, so much of the justification of the curriculum is.

Speaker 2: I think we need to communicate that through the themes that we use to deliver the material. I don't think we're there yet because we really need to get a handle on what we need to deliver before we can really start to make it more relevant is probably the wrong phrase, but it might be right at the moment. By make it more relevant I mean give them some more ownership of what they're learning, not just as a skill, but in a context that they can relate to. Who wants to sit and flip a coin? Where could we use random instead and just a simple binary choice, is something that they would directly relate to. But most of them toss a coin so that will do. Do you know what I mean? It's the same way when we're talking about encryption.

Speaker 1: Unfortunately I doubt many of them play games that use dice, either.

Speaker 2: No, not anymore. No, the dice are already thrown aren't they? Funnily enough in the games that they get to play, the RPGs they're playing, they just run and the dice is being repeatedly thrown and they just don't see that happening in that weird dodecahedron dice type scenario. That doesn't really exist much anymore either. Those kind of different shaped die that they don't get to see. Yeah, so I don't know.

Speaker 1: It's an interesting one. Especially with the sets and thinking about the different projects for the different sets, the difference between the encryption project and the coin flip project. Those are very different framings of similar ... There's similar skills needed for both. They're not the same, obviously.

Speaker 2: It's the content, the context of the ... The ability to grasp the content is slightly harder. But having said that, if we told the lower set that are doing coin flips that actually they wouldn't be sending text messages, emails, or using Facebook or Twitter without this tool, then they're like, "Oh!" But it wouldn't mean to say they understand it anymore.

Speaker 1: No, absolutely not.

Speaker 2: So it wouldn't really help too much. Yeah. Go on. I can just see it will ... I mean P– would agree with this. I said we'd need to listen to each other's answers, it would be interesting to see, reverse it. P– would agree with you, this will be a forever evolving ... It always has been. We've never stood still. I could say ... When P– turned up. It was at least more like a team than it ever was in the past. It was me, but now it's we which is great. But I think that that's ... As I said, especially increasingly as that push is coming from here. So what in five years time might have been ... What we're seeing now is something fairly rudimentary you could argue, if they keep pushing down from the top we could be ending up with having to deliver something-

Speaker 1: One of the nice things about my research, but also the challenges of it is that there is a certain sense to say this is not the right time to be ... The curriculum is way too new to be even asking the questions about [crosstalk 00:24:35].

Speaker 2: Yeah, no, it's a fair point. So are you going to revisit or not?

Speaker 1: I don't know. Maybe come back in a few years.

Speaker 2: If I'm still here, shoot me. Which is I love it here.

Speaker 1: 17 years and counting.

Speaker 2: Mm-hmm (affirmative). It's not a bad thing. But it is actually, but-

Speaker 1: Well, no, it's not.

Speaker 2: There's a bit of that. I would have moved by now but now I've got a little girl so I can't go anywhere.

Speaker 1: You're stuck, you're stuck here forever.

Speaker 2: Yeah, just a bit longer now.

Speaker 1: Children make you-

Speaker 2: Pay the mortgage off and then I'll think about it.

Speaker 1: Then they get into a school and you like the school or you don't like the school, but you like the school.

Speaker 2: Well they could come here with me.

Speaker 1: Then they're stuck because you think, "Well, I can't move because then they'd move schools. Moving schools is, we don't want them moving schools." Then they have their friends and [inaudible 00:25:24].

Speaker 2: Yup. I could bring them here with me. Saves on a lot of ... So many other bits of hassle. But then they become what I was, which was an after school child.

Speaker 1: Are your parents teachers?

Speaker 2: Yeah, both primary. My dad was ... I used to catch up with my mum after school, my brother and I both did. Wait for my mum.

Speaker 1: And play with the computers.

Speaker 2: No, I used to play on the piano actually.

Speaker 1: All right.

Speaker 2: Not particularly effectively but I did use to play the piano. I'm sure my brother was ... Yeah, we did that. We both wandered up. Yeah. So we lived in the middle of nowhere. In the sticks.

Speaker 1: In the sticks of [Allswater 00:26:09]? Or not Allswater, [inaudible 00:26:11] which is pretty far in the sticks.

Speaker 2: Two and a half miles out. Show you the house, Google Maps it for you, you'd realise where I grew up. Air rifles and running around fields being a lunatic. Wrong interview, right.

Speaker 1: Tell me about your childhood.

Speaker 2: Yeah.

Speaker 1: But not like at the doctor.

Speaker 2: Yeah. Most of it was spent in barns. Running around in beautiful green fields.

Speaker 1: You lived 20 miles down the road and ...

Speaker 2: Actually kind of the reverse. I lived 20 miles away from the inverse ... I live in the inverse of what I grew up in. Not literally, obviously, because I don't live in London which would be the complete inverse.

Speaker 1: But you look across the bay?

Speaker 2: Yeah, yeah. I could show you now, I've got time on me, got a minute.

Speaker 1: Is there anything I haven't asked? Anything-

Speaker 2: I don't know really. I don't tend to think about this stuff, so.

Speaker 1: That's all right.

Speaker 2: You've done all the thinking with all the questions, so.

Speaker 1: Yeah, no that's fine. It's just sometimes there's that one ... I always ask people what I should have asked because I figure sometimes you're sat there and somebody's asking questions and you're dying to give an answer and they don't ask the right question.

Speaker 2: I suppose the question you could ask could be is how long do I think this version, this incarnation will actually last.

Speaker 1: Okay. How long do you think this version will last?

Speaker 2: Don't know. It may be another 15 years. It may be ... [inaudible 00:27:39] obviously the content is going to evolve quite considerably, but at what stage will it backfire on itself and somebody with a different view of things might turn around and say, "Well, actually ..." Or somebody might actually turn around and go, "You know all those kids that we wanted to be really good at cybersecurity?" "Yeah?" "Well now it's backfiring on us because [crosstalk 00:27:55] cyber criminals. Maybe we should just oppress them all and just give them all basic computing or ICT skills." "Good call, let's do that." So it might be that. It might be society that decides that it was the wrong move. By society I don't mean the people, clearly.

Speaker 1: No, obviously.

Speaker 2: The real-

Speaker 1: That would be the absolute, you know ...

Speaker 2: You're in dystopian future.

Speaker 1: So much of that was that the call for more ICT, more computer graduates, more people with computing skills. I noticed at one point that the [inaudible 00:28:30] website was one of their funders was a big bank, Morgan Stanley is one of their ...

Speaker 2: Yeah, see, my brother works for one of the big banks at the moment.

Speaker 1: So they're ... And they're funding it because they need people with the skills.

Speaker 2: They need systems analysts probably. My mad brother's degree was systems analysis. I don't think that exists anymore as a degree. I don't know whether it does.

Speaker 1: But they need people with good computing skills, but it would be very funny if we become the Russia of the future with computing.

Speaker 2: Well can you get those good computing skills from good mathematics? From higher level math, my math, and things like that? Can you not ... Would they not ...

Speaker 1: Does it need to ... Do you need the language or do you need the logic, the mathematical logic? Or do you need to understand how to apply computing in creative ways?

Speaker 2: I think to be innovative I would imagine yeah you probably do.

Speaker 1: That's something I think is sometimes missed when we think of computational thinking as problem solving, which I think is the best way of thinking of it. It can sometimes miss the creative aspect of ...

Speaker 2: Okay, let's not ... Just to reinforce the value of the Micro:bit. Now I've got over the stresses of doing that evening. Last night, coding club, we got a little wireless network together. "What can we do with it then? What should we do?" "Well let's get it to control a [inaudible 00:29:52]." "All right, fair enough, let's do that, let's get that to happen." As I said to you last night as I was driving home I thought about, "Oh I tell you what, why don't I see if we can get them develop a little text messaging app that will just send it to a group of people and we'll give that a shot." So I don't know how far the lady's got at lunchtime, but Melissa, the shorter of the two [crosstalk 00:30:10]