RUNAS RADIO

http://www.runasradio.com

RunAs Radio is a weekly Internet Audio Talk Show for IT Professionals working with Microsoft products. The full range of IT topics is covered from a Microsoft-centric viewpoint.

Text Transcript of Show #168
(Transcription services provided by PWOP Productions)

Richard Campbell on the Differences Between IT and Dev!
June 14, 2010

[Music]

Brandon Wenn: From runasradio.com, you're listening to RunAs Radio, the Internet audio talk show for IT professionals with Richard Campbell and Greg Hughes. This is Brandon Wenn, announcing show #168, with guest Carl Franklin, recorded Wednesday, July 14, 2010. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com. You can follow the boys on Twitter at twitter.com/runasradio.

Carl Franklin: Hey, this is Richard Campbell. Faked you out. It's Carl Franklin. I'm interviewing Richard Campbell on his own show. Hey, Richard.

Richard Campbell: Hi, buddy.

Carl Franklin: How are you doing?

Richard Campbell: This is very odd.

Carl Franklin: Well, you know, we do this at PWOP Studios once in a while. We interview each other because we have things to say.

Richard Campbell: Yes, sometimes we're actually allowed to say to say things. I found out when you're the host, it's never about you and you don't want to talk too much.

Carl Franklin: Of course. Yeah, I mean I certainly don't and that's why -- you know, one of our best .NET Rocks! shows was when I interviewed you for show 300.

Richard Campbell: I remember, yeah. We had a lot of fun with that.

Carl Franklin: Lots of great stories.

Richard Campbell: Yeah. Those were all developer stories though.

Carl Franklin: They were developer stories. Well, some IT stories. The story about you fixing the computer is priceless though.

Richard Campbell: That is a rather IT-ish story.

Carl Franklin: It's an IT-ish story. Have you ever told it on the show?

Richard Campbell: I don't think I have ever had actually.

Carl Franklin: Maybe we should start with that.

Richard Campbell: You wanted to just start with right into that?

Carl Franklin: Well, I mean why should I have all the fun. Your listeners want to hear that story too I'm sure.

Richard Campbell: So this was a stock brokerage company and this has literally turn the way back knob to like 1987, so just dating myself. I was a well-known consultant on computer systems in Vancouver then. This is one of the few times in my life I actually did work in Vancouver and I got a call from a competitor, from a fellow consultant. You know, we're all kind of friendly...

Carl Franklin: Right.

Richard Campbell: But we are competing for the same customers and she says to me, "I hear you're the guy who could crack the tough nuts, like you really know your way around hardware and you could solve the hard problems. I have been working on this problem for weeks now and I can't seem to solve it. Are you up to a challenge?" Well, yeah, of course. Right? Like I'm all powerful, why wouldn't I?

Carl Franklin: I'm immortal, you know, so therefore...

Richard Campbell: And so therefore, well, I was very flattered actually to get a call from a competitor in the first place.

Carl Franklin: Yeah.

Richard Campbell: And I think at least give it a shot. I'm like, "Oh, what could it be?"

Carl Franklin: Right.

Richard Campbell: So this is in the era of the XT, the PC XT, so still '80, '86.

Carl Franklin: I remember what I was doing. I was in my apartment in Orlando, Florida or Altamonte Springs, Florida, hacking QuickBASIC on my XT.

Richard Campbell: There you go, yeah. That's exactly the era. And this particular computer -- so this is a stock brokerage company, whole floor full of stockbrokers and this computer that's having the problem is an XT class machine but it has got a serial model attached to it that is taking the feed from the stock exchange and making it available to all of the other computers that are in the office.

Carl Franklin: And needless to say this computer is probably making them a ton of money.

Richard Campbell: Well, and all it's always doing is distributing a data but the real thing is when it's not working, none of the brokers are getting their quotes.

Carl Franklin: Right.

Richard Campbell: So they've basically registered with this machine all of the stocks they want to get current pricing on and they're getting essentially a real-time feed via this machine.

Carl Franklin: Pretty critical piece of the puzzle.

Richard Campbell: It was a bad thing and it was hanging, just sort of dying on them and she thought it has something to do with the serial port and we looked at the modem and poked around the machine. It's all DOS 3.1 sort of that era of computers. I thought about it for a while and we talked about all the things she has tried and I said, "Yeah, I think I can fix it." She said, "Really? Will it take long?" I said no, I will probably do it in about a day. She goes, "Great. How much?" I said two grand. She's like, "Ooh. That's a lot of money." I'm like, "Yeah. You know, I'm sure I can get it fixed for two grand." And she said, "Well, I have to go back to the customer and talk about that." I said, "That's fine. No problem." And I left, I just walk away. No big deal. You know, if she pays me, she pays me. So a week goes by.

Carl Franklin: Yeah.

Richard Campbell: She calls. She says, "I've tried everything. I still can't get it to work. There's a check for $2,000 waiting for you if you will come in and fix it right now." And so what did I do? I brought a new computer in. I showed up with a new computer, I physically move the hard drive over, I started up the new machine, got it all up and running, it worked perfect. No problem. You know, back then it wasn't a big deal to switch hardware. I brought a different computer in and she went, "So, what was the problem?" "I don't know. I don't care. I can't believe you spent weeks on this. Who cares what the problem is. Switch out the machine. What are you thinking? Are you insane?"

Carl Franklin: There's always a solution to every problem.

Richard Campbell: You know what the problem really is? We like our jobs too much.

Carl Franklin: Yeah, that's right.

Richard Campbell: And this is totally independent of whether you're a developer or an IT person, we get hooked on the problem. It's like an addiction. I want to solve the problem, not that I actually want to serve my customer. I want to solve the problem more than I want to serve the customer.

Carl Franklin: We get hooked on the process. The stuff that we do we like to sit and do it because it tickles our brain and we like it.

Richard Campbell: Oh, absolutely. Yeah. And just gnawing away on the problem. You know, of course I don't know that I'm that brilliant in as much as she's explaining all these things and I couldn't think of what else to do but I could think of a way to get it up and running again. So I think that one of the points was that I was not as invested in the problem as she was and so I just didn't have that same concern that, you know, you're too busy gnawing on the problem to actually just solve it and make it go away.

Carl Franklin: Yeah. What a great story.

Richard Campbell: Ah, you know, we all have those moments.

Carl Franklin: Heck if I know.

Richard Campbell: I don't know. I don't care.

Carl Franklin: Yeah. So what are we going to talk about on your show? What topics haven't you covered that I can ask you questions about?

Richard Campbell: You know we normally drill into very specific IT technologies. Right?

Carl Franklin: Yeah.

Richard Campbell: Different features and so forth, but one of the conversations I often have with others when we're at a show like TechEd is, because I switch hat, I spent some time in development, I spent some time in IT.

Carl Franklin: Yeah.

Richard Campbell: And people get confused by that. They're like, well, is that even possible? And I sort of develop a list of the differences between the personalities or sort of the roles because I have found that I can't do both at the same day.

Carl Franklin: Yeah. You really have to -- it's really that different, huh.

Richard Campbell: It's really a different way of thinking about problems without a doubt. I fell into this, I mean one of the advantages is I've been in the business so long that I build my own hardware and back then you have to program so programming is a natural extension of the whole thing, but I really got into that mix of life in the VB days so in the era of the Windows 3.0 and VB 3.0 and that particular era was when I like the computer guy for a marketing company. It was one of the few times I've had a "real job."

Carl Franklin: Yeah.

Richard Campbell: They call me an IT Director but I was a one man band most of the time. Occasionally I have people helping me and one of the things I discovered is you can only wear one hat a day. So today is the day to do software development, I really can't run around fixing printers. I need to develop today unless it's an absolute emergency which means I now threw away a day of development. I used to measure my productivity in interruptions per hour.

Carl Franklin: Development is just physically different. I mean, you're sitting at a machine working with your brain and these abstract concepts.

Richard Campbell: Yes.

Carl Franklin: IT typically involves a lot of running around.

Richard Campbell: Yeah.

Carl Franklin: And getting down in your hands and knees and unscrewing, and screwing, and testing.

Richard Campbell: Yeah and constantly thinking about what is the real problem here. It's a lot more diagnostic work.

Carl Franklin: Right.

Richard Campbell: In some ways, software development can be like that when you try to fix a bug.

Carl Franklin: Right.

Richard Campbell: That's why I came up with this measurement of interruptions per hour because I found there's a productive IT person if I was serving my constituents here, serving the employees of the company.

Carl Franklin: Yeah.

Richard Campbell: They will ask me for something two or three times an hour. It was normal. My interruption rate was between three and four per hour. But to be productive as a developer, I couldn't be interrupted for four hours.

Carl Franklin: Right.

Richard Campbell: My interruption rate had to be 0.25.

Carl Franklin: Yeah.

Richard Campbell: And that was the whole -- and so I started having a sign on my door that basically said this is my current interruption rate. So whenever someone comes into my office, I would immediately change the number before I speak to them because I've already been interrupted. It's been down but it was a way to non-confrontationally create this reality of you have impacted me. Understand. It was mostly for my boss but it worked like a charm, this idea.

Carl Franklin: Yeah.

Richard Campbell: He was also the guy who would say, "All right, I need this project done today. We need them by 2:00." It's the morning. Then he calls every hour. "Is it done yet? Is it done yet? Is it done yet?" And so I kept adding half hour to delivery time.

Carl Franklin: Right.

Richard Campbell: I was like, "It will be on at 2:30." "You said 2:00." "You phoned again, and if you phoned again it will be done at 3:00"

Carl Franklin: Yeah. Well, and what you're really doing is conditioning them not to bug you when you're developing.

Richard Campbell: Yeah.

Carl Franklin: It's really important that you have that free time. We've talked about this a little bit on the .NET Rocks! Live Weekend about when do you code best? Do you code best in the middle of the night? Are you a morning person? You know, I tried the whole working at home in writing code thing and that just doesn't work because it doesn't matter what day, night, whenever, if your wife can't sleep she's going to come downstairs and talk to you.

Richard Campbell: Without a doubt.

Carl Franklin: If your kid is sick, you're checking on him all the time.

Richard Campbell: And I think it's is the one of the strongest distinctions. So there's a strong case to be made for isolating developers from the rest of an organization where IT needs to be accessible to the organization for the most part.

Carl Franklin: Interesting.

Richard Campbell: They're literally -- in one office that I've worked with, we put all the developers on a different floor.

Carl Franklin: Yeah.

Richard Campbell: Literally we wanted it. Not that they had to be isolated from each other, they still had cubes and they use headphones and so forth but they needed to be away from the rest of the org so they could focus.

Carl Franklin: Interesting.

Richard Campbell: Or they wouldn't get any work done.

Carl Franklin: Yeah, that's true. I remember one company I worked at, the big deal was the day everybody got offices.

Richard Campbell: Right.

Carl Franklin: Sort of like in little cubby desks or whatever and then it was just difficult because people would walk around and talk, like the CEO would come down and show people, oh, here are the developers in their natural habitat, you know, and throw us pizza and walk out again. You know, it's clearly distracting and then we distract each other enough.

Richard Campbell: Yeah. But I also worry about developers being in their own offices because they can thrash for days. They could not actually -- if they get stuck on a problem, they stay stuck on that problem.

Carl Franklin: That's your manager's fault really.

Richard Campbell: I tend to agree. But you know, our friend Stephen Forte, when he was running Zagat he put all of his developers in one big room with cubes but it was isolated from the rest of the organization but at least the other developers could see each other so if somebody is starting to thrash he can get help but nobody else can help that guy and you don't want to interrupt him if he's on a flow. There's nothing worse than actually having all those threads picked up and writing good code and the phone rings.

Carl Franklin: Yeah.

Richard Campbell: You know, anything happens. You want to get that whole brain dump done before you move on.

Carl Franklin: Right, right. Yeah. So another difference, maybe an IT person should be on Facebook and reading tweets and all that stuff, whereas, a developer really just can't do any of that. Put your email away, put your Facebook away, put your cell phone -- turn your cell phone off, you're going to be here for a couple of hours and that's it.

Richard Campbell: Well, and I find that my magic block has been the four-hour block that literally I have to turn it all off or you don't get anything done.

Carl Franklin: Right.

Richard Campbell: You know that's not to say that an IT doesn't need to think. But generally, they're thinking about diagnostic problems.

Carl Franklin: Right.

Richard Campbell: They're thinking about, you know, how do we not make this worse. The funny thing about IT is that in a crisis, that's when you think the most.

Carl Franklin: Right.

Richard Campbell: You actually stop. Rule #1: Don't make it worse.

Carl Franklin: Right.

Richard Campbell: Right?

Carl Franklin: Yeah.

Richard Campbell: If something is broken, you're not quite sure what is broken, don't go changing things until you isolated the problem really.

Carl Franklin: It all comes down to testing and sort of debugging hardware, but isolating things and starting, you know, simplifying things. Well, first thing I do when my computer doesn't work and I don't know what's wrong, it's I take all the cards out of the computer.

Richard Campbell: Yeah, at that point you literally get as few things necessary to make the machine boot.

Carl Franklin: Right. Unplug the USB keyboard, get a regular keyboard or a regular mouse, unplug everything. Get your baseline video just -- even take out the video card and put in a little crappy one that you know works.

Richard Campbell: Yeah. But you know, same thing with an Exchange Server. Nobody can get mail. You sort of go to the basics so Exchange Server up. If it was off, it's a good thing you can't get mail. Is the internet connection up? So many times it's really the equivalent of are you plugged in.

Carl Franklin: Yeah.

Richard Campbell: You know, you go and do all those checks before you change anything and it's just that patience to work through the problem that way. The challenge here with IT I think in some ways compared to development is that you can always make things worse where in development, you rarely. You can usually revert a version. Reverting a version of hardware is much harder.

Carl Franklin: Yeah, that's true.

Richard Campbell: You can't really roll back that way. You have to be very careful about what you go forward with.

Carl Franklin: Very true. Do you think that -- and now we're getting into sort of the complexity of all the gear that we have in an office. Do you think that's one of the reasons toward pushing towards the Cloud and virtualization?

Richard Campbell: I think so. I also think that you're seeing that this concept of refactoring your infrastructure, that's really what we're talking about. You know, if you look at it from a developer's point of view where we take a chunk of code and we reorganize it to be more efficient and more scalable, Cloud is very much this I have this hodge-podge of pieces, but now I've come to appreciate that these things are important. They're all equally important whether they may be my file services, or my email, or my web and we want a certain level of service from them and rather than creating those pieces myself, can I push pieces of them off to the Cloud.

Carl Franklin: Yeah, interesting.

Richard Campbell: You know, on stage, when I've done talks along these lines, because I always seem to straddle a line between dev and IT, I'd say "The problem with being an IT guy, you developers, is if I do my job perfectly you can't tell."

Carl Franklin: Yeah, that's true.

Richard Campbell: I'm totally invisible. I get a C. The best I can get is a C. I break stuff, I get an F.

Carl Franklin: Yeah. For talking on a regular basis, there's something wrong.

Richard Campbell: Yeah. You know, that's the reality of being tech support, it's you really want to sign off the call with "Hope you never have to call me again."

Carl Franklin: Right.

Richard Campbell: Do your job right and nobody can tell. I got lonely because I was good at my job so I turn the server off because people phone every time. I guarantee it. You shut the server off, the phone rings immediately. Look, they love me again.

Carl Franklin: I want to talk to somebody. Buzz.

Richard Campbell: I want to talk to somebody. Just turn your server off. It works like a charm. But compare that to a developer where every time a developer ships a new feature, people love it. There's you're A. You know, you get you're A.

Carl Franklin: So who are we talking to in the show, developers or IT people?

Richard Campbell: We're mostly talking to IT folks.

Carl Franklin: So just trying to get them to understand the difference between you and me.

Richard Campbell: Yeah, and a feel for that particular set of problems.

Carl Franklin: Yeah.

Richard Campbell: Because devs are naturally exploratory. That's their job. You're trying to invent new things. IT folks are naturally conservative because when you change every customer name to John Smith, it's my fault that I let you do that.

Carl Franklin: Right.

Richard Campbell: I'm not just being grumpy because it's fun.

Carl Franklin: Yeah. So if you want to have a better relationship with your developers, maybe this is a good segue here if you're an IT guy, maybe you don't care about having a good relationship with the developers but is it important?

Richard Campbell: You know, there are certain jobs that I've dealt with where when those two teams work well together, everybody's job gets way easier.

Carl Franklin: Now, is that because the IT person can, if they're on good terms with the developers, can say "You know, I could use a little shell script here on something that I can't figure out. Can you write that for me?"

Richard Campbell: You know, that's one element of it and I'm surprised at how many IT folks these days, because of PowerShell, are starting to learn .NET.

Carl Franklin: Yeah.

Richard Campbell: Because that's a lot of power. But the bigger thing here is you know one of the most terrifying things as an IT guy is a critical application that's been written in-house is down.

Carl Franklin: Yeah.

Richard Campbell: And you really don't know why.

Carl Franklin: Right.

Richard Campbell: All the servers that it's supposed to run on are running. The internet connection is working. Literally, because you have no view into the code, you don't understand why that's broken and you're sort of limited to rebooting servers or restarting services and trying to get a feel for what isn't working here.

Carl Franklin: Right.

Richard Campbell: And typically, in-house apps don't write good errors to the event log so there's not a lot of instrumentation for the IT person to really understand what's going on and that's one of those moments where having a dev guy that you could describe a symptom to during a firefight that he might go "Ah, yeah. Yeah, I know what that is. That could be this bug here."

Carl Franklin: Right.

Richard Campbell: We rolled over to another database and it turned out the two database structures weren't exactly the same and this table allowed nulls and this table didn't allow nulls. All we really know is when we switch over it's stopped working after an hour when a new order got entered or something like, you know. And so now you want to roll back and does not make sense that it was quicker to just fix it another way. So there's that whole conversation but where the IT/Dev relation really gets close is when you start doing serious performance to it.

Carl Franklin: Yeah, that's right because you need the developers to cooperate with you in terms of instrumentation and things like that.

Richard Campbell: Well, and typically the IT guys, we have pretty good metrics. I mean, we know how to use performance monitor. We know how to instrument up our websites. We can tell you what are the busiest pages and so forth. This is just a question of how you make those things faster and some things IT can do and some things dev guys can do.

Carl Franklin: Yeah.

Richard Campbell: And so when you can actually have a non-confrontational meeting between those two groups, the crashed website is a classic.

Carl Franklin: Have you talked on your show about maybe some of the human side of dealing with developers? How can an IT person who has a problem with a particular developer get to that place where everybody is happy without stepping on toes?

Richard Campbell: I think it's always a challenge one way or the other. It's really not a topic I think we've ever broached before, but it's a recognition that developers do like to solve problems. That's one thing we all have in common, it's that you've got to know all the problems. The challenge is structuring the problem in a way that the developer can grab on to. You tell a story too, right, where your customer calls up and says the app doesn't work. Fix the app.

Carl Franklin: Right.

Richard Campbell: And so what doesn't work? He was, "It just doesn't work."

Carl Franklin: Yeah. "I put in my username and password and it blows up."

Richard Campbell: Yeah.

Carl Franklin: Oh, okay. "What did it say? Oh, give me some error. Well, what was it?" Yeah.

Richard Campbell: So, you know, I think that's part of -- although as IT guys we can fall into metrics that devs have never seen. A lot of the devs don't know their way around PerfMon. They don't really understand, you know, hey, at this level, at this request per second, it doesn't do anything. Or this is where it crashes, or dealing with the deadlock, those kinds of things. I also think that it's useful for IT folks to know where they can push back on developers. The classic one is that developers avoid writing error-handling code.

Carl Franklin: Right.

Richard Campbell: And they really don't need to. The number of times I've been brought into an organization to resolve something like a deadlock, which is a synergy between a dev problem and an IT problem...

Carl Franklin: Yeah.

Richard Campbell: It's showing up only in production.

Carl Franklin: A deadlock meaning a thread, a threading lock where...

Richard Campbell: Well, there are all kinds of different sorts of deadlock. I mean, you have the deadlocks in database. You also have threading deadlocks but they're essentially the same thing. I have two different worker processes of some kind that need two resources and each one has one of them and each one needs the other one and they have no way to resolve that. It's a logical locking point. We're stuck

Carl Franklin: Right.

Richard Campbell: In some cases, for instance in SQL Server, SQL Server usually pick a loser, fail that transaction.

Carl Franklin: Right.

Richard Campbell: The other one goes on to finish without even knowing it killed somebody along the way.

Carl Franklin: With threading deadlocks, that can happen one in a million times.

Richard Campbell: Right. Unreproducible.

Carl Franklin: Unreproducible, yeah.

Richard Campbell: Spins the machine into orbit where nothing actually works, and then in the case of IIS you might have the worker process where you start and the problem goes away.

Carl Franklin: Yeah.

Richard Campbell: And you didn't actually fix it. You just killed it essentially.

Carl Franklin: Right.

Richard Campbell: Or worse case scenario. You have to blue-screen in your machine which again just comes back up. If the guy wasn't watching at that time, he may not even realize it blue-screened.

Carl Franklin: You know, you said something a little bit ago that made me think of a nice ploy where you said that developers love to solve problems, like if you come to a developer and say as an IT guy, "I found your problem. Here it is. Here is a list of steps that you need to do to fix the problem," they will be resistive to that, whereas, if you say "Can I talk to you for a minute? Man, I got this problem..." You know, and you've figured it out. "But I got this problem..." and you lead them to the conclusion and then let them figure it out like it's almost like the 50's housewife like Laura Petrie.

Richard Campbell: Yeah.

Carl Franklin: She's figured it all out but she wants to let Rob come up with the...

Richard Campbell: She'll let Rob come with it.

Carl Franklin: Yeah.

Richard Campbell: I don't even know as IT guys we can necessarily solve it. Usually we get to a point where we're absolutely convinced it's the code. I mean, that's the battle.

Carl Franklin: Right. And we're always finger-pointing.

Richard Campbell: I've sat in that. It's the code. No, it's the infrastructure. No, it's the code. So I think there's a very delicate point where you can bring that to a dev and say, "Here's all the things I've tried and this is the behavior I'm getting and I don't think it's the right behavior. What do you think?"

Carl Franklin: Right.

Richard Campbell: And just sort of get them into this, huh, you know, what could that be? I worked with a developer. I was the IT guy in this particular role and he always was convinced that it was the infrastructure or it was the database. It's never his code.

Carl Franklin: Right.

Richard Campbell: And over the phone and over IM which is a disaster because it was just "No, it's not me. No, it can't be it. No, no, no, no."

Carl Franklin: Yeah.

Richard Campbell: And my best solution for him always was lunch.

Carl Franklin: You know, believe it or not, I went to an audio production school, an audio engineering production school, Full Sail, way back in the '80s and I had this production class, it was great. Jon Phelps was the teacher and he came in and he said "If you learn nothing in this class, learn the power of these two words: Let's eat."

Richard Campbell: Yeah.

Carl Franklin: Let's eat. You're a producer; your job is to prevent the downward spiral that happens in a session.

Richard Campbell: Of not actually producing successfully.

Carl Franklin: Yeah. Well, in a recording session, it's all about feeding the egos of the musicians.

Richard Campbell: Right.

Carl Franklin: Because that's when they're most creative. I'm not sure how this relates to a Scrum let's say, or a meeting in which you're trying to solve a problem but there's much power in when things start to go wrong let's go eat, go share a meal and then you come back with a fresh mind, you're a little bit more on a personal level and it can do wonders. There are stories of great sessions in music that have been saved with lunch.

Richard Campbell: Yeah. I totally agree. There are two elements to that that I think are important and

this is dealing with the high tech world more than anything. First is if you have more than a couple of negative interaction with someone in a given communication medium, change the medium. Typically escalate it. So if I get a snotty IM or at least another bad IM, maybe write an email. If I get a bad email or a long -- as soon as I see a long email, I'm like stop.

Carl Franklin: Right.

Richard Campbell: Go to a phone call and if the phone call isn't working either, have lunch.

Carl Franklin: Yeah.

Richard Campbell: But always escalate in the level of content. But I do agree that lunch is magical.

Carl Franklin: It really is.

Richard Campbell: There's a social commitment to breaking bread together.

Carl Franklin: Yup.

Richard Campbell: So the funny part with that particular story was we got to the point where as soon as we argued about anything, we'd say "Barbeque." The conversation ends to where we're having barbeque. We go get barbeque.

Carl Franklin: Yeah.

Richard Campbell: And that would simply resolve it. I think you're generally more open, you bring your facts with you. "Here's the data I've got that shows those problems from this side. Here are the facts that you've got from your side." And he's like, "What if it's this."

Carl Franklin: Right.

Richard Campbell: The funny part is we were then in a position where we couldn't actually solve the problem because we're at lunch.

Carl Franklin: Yeah.

Richard Campbell: We now have to then go off and solve the problem after that. And I was dealing with very distributed teams in that particular organization so we weren't even going off to that same place. He then goes back to his office and I go back to my office and now we're communicating on IM but the tone is different.

Carl Franklin: How do you guys in IT deal with each other? I mean, there's clearly a hierarchy in IT, whereas developers, you have like teams of people. I even noticed this in small groups, in small companies where you've gotten to maybe 10 developers but you typically have one IT kingpin who knows all the IP addresses of every device everywhere, who's got all the answers, who's like the go to guy and then he got guys working for him. Don't you find that's true? Does that scale?

Richard Campbell: Typically you have pieces like that. You know, everybody has a specialty and I think this is true in development too. You got the one guy who's got the knack for Silverlight, you got the one guy that's got the knack for keeping the Exchange Server alive.

Carl Franklin: Yeah.

Richard Campbell: And that's sort of people defer to his opinion around Exchange. The guy who knows all the IP addresses is a useful guy, but he's probably a Cisco head like his brain is full of IOS and it's got not a lot of space for anything else. Keeping the firewalls configured right, keeping the routing working right, that's what he focuses on and a for normal size org or a decent size org that's very a valuable guy. But we're also trying across stream too. You usually want that guy to be able to take a vacation.

Carl Franklin: Yeah.

Richard Campbell: And so you try and have somebody else who knows the way around the infrastructure. It always worries me when people are keeping stuff in their head especially in the IT space because it's very “write downable.“

Carl Franklin: Yeah. I mean, I run into people like this all the time like they have configurations in their head, they've got the way things work and maybe it's a job security thing, maybe it's an ego thing but I know several people like this and most people pull it off just because, let's face it, if you can keep that information in your head, it's so much faster than having it on a computer, having it on a piece of paper, having it in a filing cabinet, but you reach a point of no return where you can't just do it all and then you need more people, you have to clone yourself.

Richard Campbell: Yeah. Well, and you also get back to this you eventually can no longer do any work because you're constantly retrieving information for others.

Carl Franklin: Yeah.

Richard Campbell: Your productivity goes down because your interruption rate goes so high because what's the IP of this, how do I get into this, can you give me access to that, like that's all you do all day and that eventually gets old if you've actually got other things you need to get done.

Carl Franklin: One of the smartest guys I know used to run IT for a company in California that I

worked for. If he's listening right now, send me a message and tell me you got this because you know who you are. We were sitting around, and he was from Texas, and we're sitting around out back and you know Texans sort of have -- some of them have sort of a redneck thing they play with but it doesn't matter how smart you are, there always be a redneck. So he says, "Man, you know our family, they're so redneck." Remember, this is one of the smartest guys I know. He says, "Thanksgiving, after the meal my brothers and I would sit around the table and we'll look at each other and say: You done eating? Yup. Wanna fight?"

Richard Campbell: Nice.

Carl Franklin: But he was an IT genius.

Richard Campbell: That's funny.

Carl Franklin: Well, anyway.

Richard Campbell: There you go. You know, there's the social aspect of this, as well as the work aspect. I do like the fact that the commonality of all this is that there are issues to solve a problem and a fascination with technology.

Carl Franklin: Yeah.

Richard Campbell: None of us can resist a new gadget.

Carl Franklin: Especially you.

Richard Campbell: But, you know, there's a difference between the gadgets that are on my work bench and the gadgets that are integrated into my house.

Carl Franklin: Yeah.

Richard Campbell: You do exercise things and figure out is this reliable enough for me to leave for a week, which I do fairly often.

Carl Franklin: I have a new gadget.

Richard Campbell: Do you now?

Carl Franklin: Yeah. So people who don't listen to .NET Rocks! might not know but I'm a musician and I have a band but I'm the only one who really sings. I mean, the keyboard player sings but he's sort of a solo singer like he will sing his own tune. He doesn't really do backup all that well. So I have the songs that I we've recorded where I've got backup vocal tracks and I'm also in the production engineering so I took out the multi-tracks and I just mixed these little wav files of just the background vocals to some of these songs, little background parts, and I put them all in wav files and I wrote this little program in .NET and it's a Windows application that it runs and it basically reads the contents of a directory and the song names are directory names and within those directories are wav files that are the names of the samples. It's a sample player. Now I needed a way to control this. So I bought a little musician's kind of guitar player effects pedal controller board that has a MIDI jack on the back.

Richard Campbell: Gadgetty.

Carl Franklin: Yeah. And it's got pedals. It's got like five pedals that do program changes and they send program change information over MIDI. So I had written before a MIDI library that wraps the low level musical instrument digital interface API and Windows in a managed code wrapper and I dropped that on there and I get a receive event that happens whenever something sends it in so all I really got to do is examine the data and I made up a little scheme that says if I press this pedal it calls the first song, this pedal the next song, previous song, and then to cycle through all the samples and then they play the samples. So I've got five pedals. So we're having a rehearsal every night and I didn't tell any of the guys about it. So the first sample that I have in the song is just me counting off because we actually have to do the right tempo otherwise the background vocals aren't going to be in sync when we're playing live.

Richard Campbell: Right.

Carl Franklin: The whole idea is for me to get the background vocals playing while we're playing, while we're gigging. So the first one, I said let's play Drive My Car. Okay, here we go. I'll count it off, and I step on the pedal and it goes one, two, one, two, three and now comes in the drums and base players just cracking up like they're just completely losing it and I think it's over and now we're just playing and stuff and I'm singing, we get to the part where the background vocals are coming in, I stepped on the pedal, beep, beep, beep, beep, yeah, everybody just loses it.

Richard Campbell: Automation is funny.

Carl Franklin: Yeah. So that's my new toy.

Richard Campbell: That's clever.

Carl Franklin: But it goes to show you, you know, that's what we do. We're geeks, we have toys, we play, we deal with computers and...

Richard Campbell: Yeah. Generally, if I want to go into a new shop and ask for a favor, lead with a gadget.

Carl Franklin: Absolutely.

Richard Campbell: Bring a gadget with you.

Carl Franklin: Well, I guess that's about it, Richard.

Richard Campbell: I think we're about out of time, my friend. Thanks so much for hosting my show.

Carl Franklin: Oh, it's good to talk to IT people about development.

Richard Campbell: We're not bad people, you know.

Carl Franklin: Great. I appreciate it.

Richard Campbell: And we'll talk to you next week on RunAs Radio.