Hanselminutes is a weekly audio talk show with noted web developer and technologist Scott Hanselman and hosted by Carl Franklin. Scott discusses utilities and tools, gives practical how-to advice, and discusses ASP.NET or Windows issues and workarounds.

Text transcript of show #72

July 5, 2007

Be a Better Developer in Six Months

Scott and Carl chat about building your own PC with a focus on the developer, rather than the gamer.

(Transcription services provided by PWOP Productions)

Our Sponsors

Lawrence Ryan: From hanselminutes.com, it's Hanselminutes, a weekly discussion with web developer and technologist, Scott Hanselman, hosted by Carl Franklin. This is Lawrence Ryan, announcing show #72, recorded Saturday, June 23, 2007. Support for Hanselminutes is provided by Telerik RadControls, the most comprehensive suite of components for Windows Forms and ASP.NET web applications, online at www.telerik.com. Support is also provided by /n software Red Carpet Subscriptions, the most comprehensive solution for adding connectivity to your .NET and ASP.NET applications with components for every major Internet protocol, online at www.nsoftware.com, and by .NET Developers Journal, the world's leading .NET developer magazine, online at www.sys-con.com. In this episode, Scott and Carl discuss ways to become a better developer.

Carl Franklin: Hi, this is Carl Franklin. You're listening to Hanselminutes from Hanselminutes.com. Scott is here, I am here, all is right with the world. Hi Scott.

Scott Hanselman: How are you, sir?

Carl Franklin: I'm well. What's up today?

Scott Hanselman: What are you going to do over the next six months to be a better developer? This is the question that has been posed.

Carl Franklin: I'm going to listen to Hanselminutes everyday that I possibly can.

Scott Hanselman: I think that will just make you a huge geek. This guy named Justice Gray blogged about he's going to read a developer book every week for the next 27 weeks and he listed all the books he's going to read with a couple of slots open for people to suggest.

Carl Franklin: When is he ever going to write code?

Scott Hanselman: I don't know, but he asked after listing all of these books that he's going to read, the trick is if you want to accomplish a goal, you write it down, which of course we've talked about before and he's talked about before, but most importantly is share it with other people, right?

Carl Franklin: Right.

Scott Hanselman: You can't just decide to quit smoking and then tell no one. So, what he's doing is he's saying, "I wanna be a better developer," and he's blogged about it, so that's his "share with someone else." Then my buddy in Canada, Bil Simser, has posted being a better developer in six months. This is kind of a blog meme, it spreads around because you're usually asked to tag people.

Carl Franklin: Tag people?

Scott Hanselman: Yeah.

Carl Franklin: Like "tag, you're it" kind of thing?

Scott Hanselman: Yeah, exactly. The way that you get these things to spread is you say, "Here are five movies I wanna see and here are five people and I wanna hear about their five movies." It's the classic pyramid scheme. In this case here though, he's saying, "Here's what I'm gonna do to be a better developer for the next six months," and in the tagging he's tagged myself and a couple of other folks like Ayende Rahien, Jim Miller, Jeffrey Palermo, Rocky Lhotka, and I'm sure this will spread all over the place, so by the time that this episode of Hanselminutes has come out, everyone on the Internet has posted what they're going to do for the next six months.

Carl Franklin: Ah, very good.

Scott Hanselman: But I thought it was a pretty darn good question and it got me thinking about that kind of stuff. We've discussed Ruby before and how this year Ruby was the language that I was going to try to teach myself in order to be a better developer. It seems that if you want to be a better developer, really, the first step is making the decision like accepting the problem. What are the 12 steps?

Carl Franklin: Yeah, that's right, the 12 steps to being a better developer.

Scott Hanselman: Yeah, first one is accepting you have a problem, right? I'm not nearly the developer that I think I am, that I would want to be.

Carl Franklin: My name is Carl Franklin and I suck as a developer.

Scott Hanselman: Hi Carl! Yeah, it's kind of like that, but Bil has a very good outline here and he talks about reading is one, reading books, and people are

always emailing me and asking me about what books they should read to become a better developer. I think that's a part of it, but for me it's less about reading books about development I'm realizing it. This is just for me. Everyone has their different learning style. For me, it's about reading other people's code because I really feel like if you want to be a good writer, like say you're writing novels or fiction, I think the way to be a good writer is to read as much as you can.

Carl Franklin: Read, read, read. Yeah.

Scott Hanselman: Exactly. So, I'm finding that I become a better developer when I read more of other people's code.

Carl Franklin: I'm totally in agreement with you 100%.

Scott Hanselman: Really?

Carl Franklin: Yes. In fact, I learned a lot in my earliest days of teaching myself programming just by reading other people's code. Good code.

Scott Hanselman: Not other people's bad code.

Carl Franklin: Not other people's bad code. You learn the bad habits from them.

Scott Hanselman: Well, I think that the trick is finding good code to read, right?

Carl Franklin: Sure.

Scott Hanselman: Certainly, in one's reading, they can read good code and bad code just as you can read good fiction and bad fiction, but if you really want to be a great writer of fiction, you're going to want to read as many good books as possible. So, I needed to fine a pile of really good code. What is good code? Is it beautiful code? Is it successful code? Is it code with a lot of coverage? I figured that code that is good is code that is used, code that a lot of people are using, code that is in a successful project.

Carl Franklin: Code that runs.

Scott Hanselman: Yeah, code that runs. This is not necessarily academically beautiful code, but it is code that is successful because people use it.

Carl Franklin: Right.

Scott Hanselman: Maybe the folks that are on the projects like Rss Bandit or dasBlog or Subtext don't think their code is very beautiful, as someone who works on dasBlog, I can assure you that when you have looked into the codebase of dasBlog, you have looked into the face of evil, but it works and it works for us and we enjoy it and we have a good time and we do some neat stuff and then we sell our souls and then we do some neat stuff and we sell our souls. One of the things that I'm going to do to make myself a better developer is spend some time reading, not just using, but reading some of the code that I think is pretty slick. I'm going to spend some time looking at the Castle Project. We talked a little bit about MonoRail…

Carl Franklin: Right.

Scott Hanselman: When we talked to the guys up at Illusion Software in Seattle there. I've decided that for the next six months, during the next six months, I'm going to spend some time looking at the whole Castle Project stack. That means MonoRail, that means the inversion of control container, and I think we're going to do a show on the next week or two on inversion of control and dependency injection.

Carl Franklin: Okay. Yeah.

Scott Hanselman: I think it's an important topic we haven't covered yet, the Windsor Container. Also, the dependency injection ObjectBuilder stuff that came in the Enterprise Library.

Carl Franklin: Right.

Scott Hanselman: I'm going to read this code. I'm going to try to understand what's going on, not just skim it, but really try to grok it, try to drink it in.

Carl Franklin: Yeah.

Scott Hanselman: I think the Enterprise Library, the stuff that Microsoft patterns and practices releases, sometimes that gets a bad rap. It's a different kind of code, right? I mean it's still good, but it has a different feel just like different writers of fiction have a different vibe.

Carl Franklin: Well, also, there is so much code there and so many different categories that you really can't consider it all as one piece.

Scott Hanselman: Yeah, definitely. If you know the guys on the team, you can tell who was in where as they try to keep things looking the same.

Carl Franklin: Right.

Scott Hanselman: I think I learned a lot from taking a look at the code within Reflector of the base class framework, looking at .NET.

Carl Franklin: Oh yeah.

Scott Hanselman: You're not necessarily reading the code because you're looking at a decompile, but you're gleaning the programmer's intent by looking at what Reflector thinks they meant and that can help in designer frameworks, but I want to sit down and look at Subtext.

Carl Franklin: Sure.

Scott Hanselman: Phil Haack's team stuff, very well thought of, and I think there's a lot of stuff that the dasBlog team can learn from Subtext and I want to check that out. Where would you think would be a good place to find a good code to read, Carl?

Carl Franklin: You mentioned them. Enterprise Library is a great place to start. Anything that's a public domain, kind of open source project is also a great thing because what's good about that is you get many eyes looking at and revising code and stuff that's open source tends to be a little more, I won't say polished, but the guys who are writing that, the guys and gals who are writing that stuff, they know that the eyes are looking at it. It tends to be more commented and it tends to be thought out a little bit more than just stuff that's slapped together that you could coerce a developer into sharing with you, something that wasn't meant for public view.

Scott Hanselman: That's an interesting perspective. I would partially agree. I would say that when you're writing a framework that people are using, like if you're writing on the Castle Project or you're writing patterns and practices, if you're writing code that is definitely going to be reused by other folks as code, I think you're right. You spend a lot more time making it clean. I think if you're writing an app like an RSS reader maybe that you'd be more willing to go nuts because it's the end user who wants the experience and beauty is a secondary goal.

Carl Franklin: Yeah. Another thing to do is to take a look at your third party tools and at some of the samples that come with those tools.

Scott Hanselman: Yeah.

Carl Franklin: Those are usually done with an eye towards "we want to do something that's happening in the main stream and we need to show the right way to do it."

Scott Hanselman: Yeah, definitely. So, I think reading is a big part of this. I'm also going to be reading a book called -- what is it called? It's something, it's F#, hang on a second. Is it Practical? It seems like every book these days, it's either Practical, Pragmatic, Professional, Powerful…

Carl Franklin: Do you want to say that again?

Scott Hanselman: No, hang on, because this is a good thing to keep in here because -- I can never remember. Is it the Professional one, the Practical one, or the Pragmatic one that is the book I want to read?

Carl Franklin: Yeah.

Scott Hanselman: Let me check my bag here.

Carl Franklin: Something that starts with P.

Scott Hanselman: I'm pulling this out of my bag. It is Foundations, ah, "Foundations of F#." I'm reading this book by Robert Pickering. It's an Apress book. These are the yellow books.

Carl Franklin: Okay.

Scott Hanselman: F# is a functional programming language, kind of based on like ML, but it lets you use the .NET framework and it has a performance kind of profile like C#, so it's really fast, but it's got functional programming perspective. It's got imperative programming perspective. It's object-oriented. I got into this because I had put up a list of books on my blog that was called "Top Language Agnostic Programming Books to Read" and one of the guys in the comments called me and he said, "This is not language agnostic. You described a bunch of object-oriented books." He said, "What about the functional programming paradigms? You know, there's a lot more out there than just if-then-else." It got me thinking and then I saw this book on F# and I said,

"Well, it's got the little # sign in it, then it must be good." Right? It's F#, so D, E, F, it must be two to three times better than C#.

Carl Franklin: I like the key of F#.

Scott Hanselman: Exactly.

Carl Franklin: Especially F# minor, quite brooding of a key.

Scott Hanselman: Well, sometimes it's difficult for me at least when you do kind of, I wouldn't say drink the Microsoft Kool-Aid, but sip it at least to go totally in the opposite direction and go download something like Haskell or Hugs and start to do that kind of work because in the back of your mind, you're thinking, "Well, how am I gonna use this in the real world?"

Carl Franklin: Right.

Scott Hanselman: This is just for me at least perhaps, but if I want to be a better developer, yes, I would like to be more well-rounded, but at the same time I need to make use of the things that I do on the side in the real world, so doing Ruby on Rails on the side, I still feel like I can use that in the real world. Learning Haskell or Squeak at this point is…

Carl Franklin: Squeak? Did you say Squeak?

Scott Hanselman: Yeah, Squeak. We'll do a show on Squeak.

Carl Franklin: What's Squeak? Just tell me what it is.

Scott Hanselman: It's a Smalltalk environment. Google first…

Carl Franklin: Okay.

Scott Hanselman: Yeah, okay.

Carl Franklin: Okay.

Scott Hanselman: Yes, and how is that useful?

Carl Franklin: Yeah.

Scott Hanselman: No, Squeak is a graphical environment. It's an open source Smalltalk-80 language that was written in Smalltalk. You go to squeak.org. You can check it out.

Carl Franklin: You say it's graphical, so you can program with…?

Scott Hanselman: Yeah, it's for kids.

Carl Franklin: Oh, it's for kids.

Scott Hanselman: Well, I wouldn't say it's for kids, but it is like an operating system, but it's an environment.

Carl Franklin: Interesting.

Scott Hanselman: Your kids might like it. Nah, of course, I'm going to get in trouble for saying it's for kids, but it's a virtual machine, it's got a lot of music and multimedia things, it's used for educational platforms.

Carl Franklin: Okay.

Scott Hanselman: It's a powerful thing. We'll talk about it.

Carl Franklin: I will check that out.

Scott Hanselman: It's not something that I would necessarily use everyday at work.

Carl Franklin: Okay.

Scott Hanselman: But I want to be a better programmer, so how do I balance that and using my time appropriately? Then I saw F# and I said, okay. Well, it's totally weird. It's totally out there. It's totally not mainstream, but it's a Microsoft research project…

Carl Franklin: Yeah.

Scott Hanselman: It's supported. I can use it in Visual Studio. It's funky enough to help me spread the wings a little bit, stretch outside of my comfort zone, but it's .NET so I can use it today. This was kind of a first step for me to go kind of way out there and learn something completely different.

Carl Franklin: Right.

Scott Hanselman: So, F# was a good compromise. Learning to write F# is going to be a part of that and the "Foundations of F#" book, I'm about halfway through it and I'm actually enjoying it very much.

Carl Franklin: Cool.

Scott Hanselman: Now, speaking, you and I have done a lot of speaking on the road and that kind of stuff. Do you think that that's a thing that one could do to be a better developer? Bil said this on his blog that he thought that that was a useful thing.

Carl Franklin: I'm not so sure. I think there are better uses of your time if you're trying to be a better developer than speaking. I would say that all of these experiences help your well-roundedness, but speaking is typically where you find several topics and you go out and you hammer them over and over and over again. There's a lot of repetition and you don't always get feedback, this was right on, this was not so right on. You don't always get -- you may have these situations where you go out and you say something that somebody could take the opposite position on, but you don't know that because it's one-sided. It's not a discussion, you know what I mean?

Scott Hanselman: Right, yeah.

Carl Franklin: I would tend to think that a discussion would be more beneficial than speaking because you don't get the feedback.

Scott Hanselman: Yeah. I would tend to agree. I think that speaking -- sometimes people want to speak at a local users group, I think those things are great; speak at a conference, and I encourage people who want to do that to do that.

Carl Franklin: Pub clubs.

Scott Hanselman: Exactly.

Carl Franklin: Those things where you get together.

Scott Hanselman: Pub clubs, nerd dinners, I find that setting up a study group at your company is a really useful thing.

Carl Franklin: Yeah.

Scott Hanselman: We started doing work with Type Mock after the mock objects framework and we tried to select a mocking objects framework for us to use here at Corillian, so the way that we did that is we had a mocking object study group that met a couple of times a week at lunch.

Carl Franklin: Speaking of meeting at lunch, we found that a lot of people are doing this with dnrTV. They're watching one day a week. They get two developers together around a projector. They have lunch and they watch an hour long conversation, presentation, demo, whatever dnrTV is, whatever you want to call it, and then after that they spend another half an hour discussing it. That's a popular thing to do.

Scott Hanselman: Yeah, I think that's a good example, so I think that rather than speaking I would say hosting discussions.

Carl Franklin: Yes.

Scott Hanselman: Talking to people, literally go to lunch with somebody. One of the things that we've also been thinking about doing we've done a little bit here, but we haven't actually formalized it with HR and the bosses is taking a bunch of our devs, a bunch of our developers out to have lunch with other developers at another company that's not in a competitive situation. It's almost like a sister city kind of thing.

Carl Franklin: Right. Right.

Scott Hanselman: So, find a company locally that's not in your space, like you do banking, they do insurance. You can all sign NDAs all day long, but sit down and compare notes. I'll go to lunch with darn near anybody. You know what I'm saying?

Carl Franklin: Sure.

Scott Hanselman: If you buy me a sandwich, I'll talk to you for as long as it takes me to eat that sandwich. Last week, I had lunch at Panic, Panic.com. Panic is a constant award-winning Macintosh developer shop.

Carl Franklin: I was going to say how did you have lunch at a website?

Scott Hanselman: I had lunch at the website.

Carl Franklin: At the store of the website.

Scott Hanselman: Well, I went to college with Steven Frank, one of the founders of Panic, and they won an Apple Design Award again this year. They win them year after year. I went down and saw their

offices down here in downtown Portland and had lunch with my buddy, Steven, who I went to college with and haven't seen in a while. I said, "Well, you got to come on, on the show and talk about Mac because some of us just don't get it."

Carl Franklin: Right.

Scott Hanselman: It would be really interesting so we'll have him on the show at some point in the next couple of weeks. That kind of stuff, having lunch with people outside of your comfort zone I find to be a really useful way to become a better developer.

Carl Franklin: So, how many Mac developers does it take to copy a file?

Scott Hanselman: I don't know. How many?

Carl Franklin: What's a file?

Scott Hanselman: Okay. How many developers does it take to copy a file on Vista?

Carl Franklin: I don't think you can actually… I don't know. How many?

Scott Hanselman: Yeah, I don't know either. It's still calculating.

Carl Franklin: Still calculating!

Scott Hanselman: Call me in an hour, I'll tell you how long it will take.

Carl Franklin: Oh, too much fun.

Scott Hanselman: I just made that one up. Anyway…

Carl Franklin: There's your [fluff] seconds for the day.

Scott Hanselman: Yeah.

Carl Franklin: So, what were we talking about? How to be a better developer!

Scott Hanselman: Yeah, going out to lunch with people that are maybe outside of your comfort zone working on things that you're not working on, understanding how they do their build servers, understanding how they do their continuous integration, do they do code coverage, how much code coverage, talking to them about their software development life cycle, do they do Agile, do they do RUP, do they do Scrummerfall? Who knows?

Carl Franklin: And we don't have to tell you that if you're in a situation where you don't have developers to bounce ideas off of readily available that listening to podcasts is another good way to be a better developer.

Scott Hanselman: Yeah.

Carl Franklin: We hear it all the time.

Scott Hanselman: It's about being a part of the conversation. You brought up a really good point. What about the single-person shop? What about the guy who's listening to this who's not driving to work because he just has to go downstairs, put his slippers on, and now he's at work?

Carl Franklin: Right, he's the only developer within 100 square miles.

Scott Hanselman: Well, I don't know about that. I guess that's -- Mike Gunderloy would be the only guy who's living in a farm with 100 acres outside of southern Washington.

Carl Franklin: Oh, I don't know. I know quite a few of them.

Scott Hanselman: Yeah. How do those guys get involved? They got to join the conversation. I think if you can go to a nerd dinner or a pub club, you got to do it. You got to get out there. I think that if I were working on my own as a single-person developer, as a one-man team, or as a remote team, being a part of the conversation is so much more important.

Carl Franklin: Yeah.

Scott Hanselman: If you can't stand up and walk over to someone's cube, you should have a lot of people on your IM list, you should have people on Skype, you should go down and sit at Starbucks. Find out where the other developers are.

Carl Franklin: Is there any such thing as a developer chat room these days? I mean I know that there are webcasts and things like that, but…

Scott Hanselman: Well, there is the IRC. Interestingly, Internet relay chat, IRC, seems to be coming back.

Carl Franklin: Really?

Scott Hanselman: Some people who are on it might say it never left, but more and more I'm seeing open source projects that have a room dedicated to their project.

Carl Franklin: Interesting.

Scott Hanselman: You'll see people who are enthusiasts about Ruby who will say, "Yeah, come visit us on the Ruby room or the Rails room on IRC," and they'll just keep that window open all the time, just kind of watching for questions, for noobs, for people who need help, and that is their virtual chat room.

Carl Franklin: Hmm. Interesting.

Scott Hanselman: Here in Portland, we've got a thing called CubeSpace, which is kind of neat. It's actually a rent-a-cube, rent-a-cubicle, so it's a big giant, open area with cubicles and Internet connectivity and phones and faxes and it's basically a drop-in and buy an office for a day.

Carl Franklin: Wow.

Scott Hanselman: Independents can do that to have a little bit more formal of an experience than Starbucks, but still be able kick ideas around and talk to other people.

Carl Franklin: Sounds good.

Scott Hanselman: Yeah, I think being part of the conversation is important and that's what it's all about, community. So, we've got reading, reading as much code as you can. Don't force yourself to read a book if you don't think it's going to…

Carl Franklin: I would agree. I think you run the risk of becoming too academic. You really got to stay in code as much as you can, I think.

Scott Hanselman: Yeah. I really like books. I have a lot of code samples. I definitely agree.

Carl Franklin: Right. Right.

Scott Hanselman: As much as I wish I were a person who could read like design patterns and truly love it and love every minute of it, I just got to see the code.

Carl Franklin: Yes.

Scott Hanselman: I think writing is good, getting it out there. Writing is part of being in the conversation.

Carl Franklin: Oh, gee. You know, we never said teaching.

Scott Hanselman: Yeah. That's a good point.

Carl Franklin: Teaching is a great way to be a developer. Doing a presentation is one thing, but actually teaching a class is a great way to sort of force yourself to learn something that you might not know.

Scott Hanselman: That is really true. You'll never be a better DataGrid expert than if you teach it for eight hours a day.

Carl Franklin: In a small class.

Scott Hanselman: Yeah.

Carl Franklin: Yeah.

Scott Hanselman: Another way of course is to get an intern. This might be a funny way to say how can I be a better developer, find an intern.

Carl Franklin: Huh.

Scott Hanselman: I've got two interns, two high school interns. Go to your local high school, go to your local community college and bring in an intern. Maybe you pay them a stipend, maybe you pay them an hourly fee, but this is going to be a fresh, kind of open mind that's going to drop into your developer environment. You're going to find out what works, what doesn't. They're going to see the real world, but it's also going to expose aspects of your process that maybe don't work.

Carl Franklin: Sure, because you get to see things through their fresh eyes.

Scott Hanselman: Yup.

Carl Franklin: Yeah.

Scott Hanselman: They're going to ask questions that are wide open and like, "Wow, I never thought about that. That does suck. I don't know why we do it that way." So, yeah, get an intern because that involves all of these things. It involves community. It involves reading code, their code, which may not be good, but it's reading code, helping them write, teach…

Carl Franklin: Yeah.

Scott Hanselman: That's a good all-around way, so I think everyone should run out and get an intern.

Carl Franklin: Awesome. Well, you know, we're just about out of time here. You got any last minute things on that list?

Scott Hanselman: No, I think we covered it. That sounds like reading, writing, speaking, community, get an intern.

Carl Franklin: Listen to podcasts.

Scott Hanselman: Listen to podcasts.

Carl Franklin: But you knew that already.

Scott Hanselman: Get on IRC, get on chat. I'm more of a chat guy than an IRC guy, but…

Carl Franklin: Yeah.

Scott Hanselman: Talk. Talk, talk, talk.

Carl Franklin: Talk and listen.

Scott Hanselman: Yup.

Carl Franklin: All right, Scott. Thanks a lot and we'll see you next week on Hanselminutes.