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 #293

November 17, 2011

Developing a mobile app for iPhone, Windows Phone 7 and Android with Toran Billups

Scott sits down with micro-ISV mobile developer Toran Billups. Toran has written, published and sold his mobile application on iPhone, Android and Windows Phone. In the process of writing BlueFlix, his Blockbuster Express movie application, he learned mobile development on three platforms. What was his experience and what does that tell us about the state of mobile development today?

(Transcription services provided by PWOP Productions)

Lawrence Ryan: From hanselminutes.com, it's Hanselminutes, a weekly discussion with web developer and technologist, Scott Hanselman. This is Lawrence Ryan, announcing show #293, recorded live Wednesday, November 9, 2011. 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. In this episode, Scott talks with Toran Billups about developing apps for iPhone, Windows Phone 7 and Android.

Scott Hanselman: Hi, this is Scott Hanselman and this is another episode of Hanselminutes, and today mobile development and the mobile development report card. I'm actually talking to Toran Billups. He's an interesting guy because he has created iPhone, Android, and Windows Phone 7applications and I thought it would be cool to talk to him and find out what the state of mobile development is today from a pretty unbiased perspective. Would you agree that you're unbiased?

Toran Billups: I'm trying to be unbiased. I came to you, Scott. You're the most unbiased, of course.

Scott Hanselman: You're too kind. You're too kind. So how did this start? You decided one day to start writing mobile apps. You needed a hobby, right?

Toran Billups: Yeah. Every developer needs a hobby outside of the corporate world. Sometime last year I was working for a software company that kind of started a small mobile team internally and a friend of mine, one of the developers on this team, had some great success with an Android application and I thought I would attempt to duplicate his success but across all the platforms, and along the way hopefully learn this new mobile development paradigm or what's exciting about mobile development.

Scott Hanselman: Did you end up going into this with the idea that you would try each of the three and end up writing a report card or was it just you wanted to do mobile and it didn't really matter which phone you used?

Toran Billups: You know, I initially started on the iPhone side thinking if I was going to fail I should fail fast and initially I felt like Objective-C, from what I had read and heard about it, might have been reach for me technically just because I don't come from a C background. A lot of my college experience and work experience was in the managed world so I thought, hey, if I'm not going to be able to do it, let's start with what everyone claims to be the hardest platform on the iPhone side. I think around this time Windows Phone 7 was just getting off the ground if I remember right. Like last year some time.

Scott Hanselman: Uh-huh.

Toran Billups: So yeah, I just kind of initially thought I would build an iPhone app and then go from there. I didn't really have plans at that time to say that at one point I want to give talks or write blog posts about how these things compare. In fact, that part didn't really come up until much later when I saw a lot of people comparing or contrasting Android and iPhone development. At that time I was already starting to do some Windows Phone and I thought, you know, nobody is talking about Windows Phone so maybe, if I get a little experience or build an app or two on Windows Phone, I could kind of contrast all three which hasn't been done that I've seen.

Scott Hanselman: Certainly not as exhaustively as I thought your post was and as fairly. Just to make sure, for folks that maybe haven't listened to the show before, I work for Microsoft but do not work for the Windows Phone team and I have no particular interest either way in Windows Phone. You don't work for Microsoft. I'm looking at your blog and you've got tags like Ubuntu, and Fitness, and Ruby, and WCF so it looks like you're a generalist. I'm a generalist that works for Microsoft on the web side. I happen to have an iPhone. What's your primary phone that you use every day?

Toran Billups: Right now it's an iPhone 4.0.

Scott Hanselman: Okay. So we're two iPhone users and I've done a couple of...

Toran Billups: But we're unbiased, right?

Scott Hanselman: We're unbiased in that we have a love/hate relationship with our iPhones, I'm sure.

Toran Billups: There you go.

Scott Hanselman: Okay. So let's start with that. Let's start with the iPhone. What's Hello World like on an iPhone?

Toran Billups: So the Hello World experience was, I tried to be nice about it mainly because I'm not an OS X user by default. I did change jobs probably a few too many times this last year, and I'm actually working on Linux right now doing Python. Prior to that I'd always done development on Windows. When I started to do some Objective-C stuff, of course the first kind of straw that breaks your back is, oh, do you have a Mac and are you running the latest version of OS X, etc? So in that part of that I kind of decided to be nice to the iPhone story and say assuming you have a Mac, the Hello World experience was actually pretty awesome because similar to the Visual Studio experience of Windows Phone you pretty much just pull down the tooling and as soon as you have the tooling up it's kind of file a new project, F5 so to speak, where you just launch the simulator and you

can actually see your app in the background so to speak. I felt it was pretty good for being a non-Mac user, too.

Scott Hanselman: So they've got a nice simulator, both iPhone and Windows Phone have a simulator/emulator. Xcode is what you use on the OS X and you just say file new project, you name it, you hit F5 or whatever the equivalent is, and it runs in a little mini iPhone on your Mac.

Toran Billups: Right. Yeah, exactly. I mean I didn't feel it would be that quick. Just from the stigma and the community of that there is a lot of effort to get your first iPhone project up, and I would say just to get a Hello World that was probably proved wrong just by me going out and doing it as a novice. I should probably throw a little disclaimer out saying that when I built this report card I wasn't trying to come at it from a "this is how the world is today on all these platforms." It was really from "I've been doing web development in C# or Java the last three, four years and what is that experience like if you want to get into mobile but you've pretty much been a web developer." So that was really the angle I came at, and Hello World for me was blatantly easy.

Scott Hanselman: Okay. But with the Android though, what's that like? What do you use to develop in an Android and what do you have to download to get started?

Toran Billups: If you go out and read the post, I was I think in retrospect a bit unfair to Android. Throughout the post, I tried to be very apples-to-apples meaning that I was not having a bias or I wasn't doing different steps. In the Hello World side, I think I went a little bit far with Android saying that if I really wanted to get Android up with minimal effort I would pretty much install the SDK and open Eclipse and at that point it is fairly file new project like. Personally I'm just not an Eclipse person so of course I want to be difficult and in turn made my life very difficult, so I decided to go the IntelliJ route which is a JetBrains product. It's the same parent company of ReSharper. You know, the experience was I would say just a little rocky because not just that Eclipse is the only way to kind of get going with Android, but if you're doing Android on Windows and you're kind of used to the out of the box F5 experience I described, you're not going to get that same experience doing Java right out of the gate. In fact, I had such a painful experience. I had a co-worker finally come over and help me out because obviously I was doing Java earlier this year and I wasn't probably accustomed to all the stuff that Java developers are on the Windows side especially, so there are a lot of steps. I kind of rated the Hello World experience poorly because the more steps there are for the developer, the more chance there is for human error. Let's be honest, that's where all the pain comes in.

Scott Hanselman: Sure.

Toran Billups: I felt like on both the iPhone and the Windows Phone side there's no real human error unless you can't figure out how to download the tooling. Once you have that installer up and running, you're pretty much done.

Scott Hanselman: So if you use Eclipse on Android Hello World, why you were a little bit unfair because you went a little slightly off, I mean off the beaten path. It sounds like Hello World is pretty straightforward on each of the phones.

Toran Billups: Right. One of the other gripes I had, especially on Windows, is if you were installing the SDK, the very first kind of prerequisite is, hey, do you have the JDK installed because you're going to be doing Java development. The weirdest bug in the installer was that if you had the JDK installed correctly, everything was fine. For some reason, the next button was just grayed out in the actual installer. So every time I would install it, I kind of look at it and go okay, maybe I messed something up. So it took me actually reinstalling the JDK a couple of different times and then full out paving my workstation before I went online and said maybe someone else is having this problem.

Scott Hanselman: That was a mistake.

Toran Billups: Exactly. And I found out there's just a bug in the SDK, but that made it painful for a newbie. Like I said I tried to come at the angle on the mobile dev side that if I've never done this stuff, how hard is it really to get going.

Scott Hanselman: Okay. So Hello World, the A's and B's if you stay on the beaten path. But when you come to the learning curve, it sounds like -- since you're kind of a .NET developer or a Java developer, it sounds like you've been doing Java since January of this year and before that .NET, you're definitely a curly brace's developer. You're familiar with curly brace related languages.

Toran Billups: Right. So you can see where the learning curve is going to take a turn towards the iPhone and kind of beat-up on the iPhone platform. As I mentioned earlier, I don't really have a C background and I was very hesitant to get into Objective-C because everyone kind of talks about of course what memory management and the pointer thing of course, if you don't know pointers you shouldn't be a programmer and all that good stuff. So I came in very afraid and spent a long time just kind of looking at the code thinking why is everything in array. When you first come to Objective-C, there are so many brackets where it feels very bracket-heavy, that the joke is just you look at it and everything is in array for two months. But I initially built my product December of last year and I kind of prototyped it for

about 30 days thinking I don't know Objective-C, I'm kind of copying and pasting this from Stack Overflow as I go. I know in the end in this 30-day sprint or a time box if you want to call it that, I'll be throwing this away. I did actually throw the product I had away because it wasn't ready for primetime. In addition to that, what's the best thing for code quality of course is hard drive failure so why not do that myself. I basically threw it all away and spent another two months kind of refining and rewriting it all. Through that process, one of the weird things I describe for a learning curve is not just how long does it take me to learn how to new up a string or new up some object in Objective-C or Java or C#, it was more how long was it before I really felt productive. Like if you kind of equate this to natural language, a Spanish teacher told me a long time ago in high school you really know a foreign language when you sit down and think in Spanish or French. You're no longer thinking in English. That's how I kind of labeled this learning curve, it's I just wanted to sit down and, boom, I just wanted to kind of be thinking in the language. But I did so much of this what I would say harmful unlearning where instead of just saying this is how I create something in Objective-C, I'm just going to run with it, I said that's how I do it in Objective-C but how would I do it in C#? Oh, that's how I would. And instead of kind of learning the language, I was learning it by proxy in a way and I would always do this left/right context switch in my head. I think Jeremy Miller talked about this a while back, that the hardest thing about learning new technologies is as you get older you have a lot of things to compare that to or unlearn.

Scott Hanselman: Right.

Toran Billups: I feel like that sort of really held me back. I'm not even 30 years old at this point, but I feel like a lot of the C#, Java, now I'm doing Python full time, a lot of those things just get in your head and you kind of have to just muscle memory it to some degree.

Scott Hanselman: In your post, you said it took two months of writing code daily for two to four hours. That seems like a long time. Do you think that that's fair, that anyone sitting down on any of these things, iPhone, Android, or Windows Phone, would need to really think about this for two solid months pretty much every day or do you think that you paid an extra tax maybe doubling the amount of time it would take because of your pre-disposition towards like a .NET language?

Toran Billups: Yeah, that's a good point. You know, a big part of it early on was I was so concerned about pointers and memory management, and every time I went to a local user group or a code camp and I talked to the person presenting on getting started with iPhone, they would always say, "Well, you have to be doing memory management. You have to understand pointers." After that spike or that initial 30-day trial that I threw it away, I said "You know what? I didn't even think about memory management and I didn't have a problem." Also, I didn't really worry about pointers. When I did that little asterisk, I just assumed that was a variable. So in my head, and this sounds kind of bad and like I said two months probably makes me sound un-hirable on your podcast, but the thing for me was just kind of throwing those fears because I feel like I had, honest to God, some kind of code fear where I was actually not very productive because I was afraid I was going to screw something up. But the two months that I really got into it two to four hours a day, I just treated everything like variables and didn't even worry about memory management until I got to the point where I was getting ready to ship it because I didn't know if I was going to make any money anyway so why would I kind of do all that optimization upfront. When I actually got ready to ship the product, I kind of look back over the code and releasing memory really wasn't as big a deal as everyone had told me. So I thought if I had gone into that without all of these negative biases against me, I might have just actually rolled with it and said, hey, it's just another programming language. I think, like you're saying, there's a bunch of stuff in my head that I had to just kind of get past before I was productive.

Scott Hanselman: Why were you so concerned about memory management? Did you have an impression of Objective-C as not being a language that did much for you? I mean doesn't it have smart pointers and some abilities to release memory? I guess, let's me put it this way. When I first did mobile development, I did it on a Palm Pilot. It was a Palm 1000. It had I think 16k of heap and 4k of stack, and memory management was a huge deal. The hardest part for me developing on a tiny, tiny mobile device in C, again much, much smaller than an iPhone, was thinking about like object scope, object lifetime. Like how long should this array be around? Am I leaking four bytes here and there? Because if I do I'm going to eventually hear about it. Do those kinds of things exist on these different platforms, on these three different mobile platforms? How much are you thinking about object lifetime?

Toran Billups: Yeah, that's really what kind of hung me up in the early days when I was building this app. It's I was so concerned about like you said leaking four bytes here and there, and come to find out this thing has got 512 megabytes of RAM on it so why am I that concerned about leaking four bytes.

Scott Hanselman: Well, but memory leaks are bad. I mean I don't want to like call you out on the podcast...

Toran Billups: Sure, sure.

Scott Hanselman: But I mean we don't want to leak memory, to be clear.

Toran Billups: Correct.

Scott Hanselman: Do the development tools help you find stuff like that and does the guidance and the prescriptive documentation for all of these platforms like tell you when you should be thinking about leaking memory or not?

Toran Billups: Yeah. So let me dig myself out of this hole here. Initially when I worked on the product, this was last December, and back then it was Xcode 3.0 which was, in my opinion, probably just the worse IDE I've ever used, and that's a bold statement. But it didn't really help you with memory management or tell you any helpful hints. Now I have downloaded Xcode 4.0 recently, and in Xcode 4.0 it's doing kind of what you're talking about which is more helpful hints. I did develop an app about a month ago for our code camp locally and found that to be a huge help because there were certainly places where I wasn't auto-releasing and it would just say, hey, you need to auto-release this. I'm like yeah, I do. So I would use the tooling in that way and it has gotten a lot better. I personally though, if you're going to do some Objective-C fulltime, I would recommend actually pulling down the latest JetBrains product called AppCode. It's essentially like ReSharper or any tool that JetBrains makes. It's kind of a great IDE. The thing I like about it is that I can transition from work and home without having to relearn a bunch of shortcuts and that's what really kind of bummed me out about Xcode. In addition to being kind of a very poor editor, it's also a whole other set of mappings you have to learn. So I'm doing Python now and the great thing is we're using PyCharm, the other JetBrains product. And then I come home and if I'm doing Objective-C, same shortcuts, same editor. If I'm using ReSharper in Visual Studio, same shortcuts as I'm using in IntelliJ.

Scott Hanselman: This episode of Hanselminutes is brought to you by Careers 2.0 and our friends at Stack Overflow. Careers 2.0 is the only career site dedicated to matching great developers with great jobs. Post your open dev positions on Careers 2.0 today and reach the entire development community, all 15 million monthly visitors on Stack Overflow. The best part is you'll receive exceptional candidates, not just the massive pile of unqualified resumes that you're used to wading through. It will save you tons of time and headaches at careers. When you're posting your job, you can even choose to receive a candidate's career profile instead of just the resume making your candidate sorting even easier. The guys at Stack Overflow believe in their product so much that if you're not incredibly happy they'll refund your money, no questions asked. To top it off, as a Hanselminutes listener, you get a free Stack Overflow T-shirt when you purchase a job listing. To get your free T-shirt, use the code Hanselminutes and check out careers.stackoverflow.com. Once again, that URL is careers.stackoverflow.com.

That's an interesting point that you bring-up because if people who are listening go check out your app at blueflixapp.com (we'll put that on the show notes), this is an application that lets you manage your Blockbuster Express movie kiosk and you have an iPhone, a Windows Phone, and an Android app so you're context switching. Really, you're a tiny ISV. You're a micro-ISV. You're kind of that dream startup, one guy in a garage with three apps all branded, all working on any device at all, and you're switching between those devices, fixing bugs and dealing with customer requests. Context switching is more than just the language and the concepts. It's also, like you're pointing out, the muscle memory. It sounds like you standardized on JetBrains tools for your muscle memory and that made you more productive. So people should take into consideration that muscle memory tax as it were.

Toran Billups: Right. I mean if you're going to be in this industry for more than a year, you're going to have to eventually be very good at it. You want to be productive and have high velocity, and especially as you mention in the micro-ISV sense, this is my time. This isn't an employer's time or if I'm learning a couple of shortcuts, so be it. This is how quickly can I get a new feature out to either beat the competition or create a whole new app that someone would pay money for. That's been, like you said, something that's reevaluated over time because it really wasn't important to me early on. I was just playing around it/ it was a hobby. But like you said, now I've got not just effort in coding but all that brand recognition work and trying to keep the name up in the internet sort of space on Twitter and get people interested in buying it or checking it out. That's a lot of work too, and so you certainly don't want to waste the little time you have kind of relearning a new editor or any other kind of context switching; that's expensive.

Scott Hanselman: Let's move on to Test-Driven Development. Because I know that I've sat down with a Windows Phone app and thought about writing it and actually stopped writing it, got frustrated and gave up for a couple of weeks because I just couldn't figure out what the right way to test one of these things is. How do you test a phone application? What's the right way to set up a unit test framework, and how does each of these phones stack up?

Toran Billups: You know, I also have a disclaimer kind of in this post because I'm still not really test driving a lot of the front-end stuff. So your question kind of begs like how are you verifying these views kind of look and that sort of stuff. I haven't really found a way to do it yet. What I kind of labeled this category as was if I'm just driving out the domain model or kind of the class library underneath, how easy is it to get that code under test? Just the actual

classes and services, whatever you want to use underneath the covers. For that, on the Windows Phone I kind of beat them up here a little bit because the Express tooling which -- Microsoft, probably like all the other vendors, is trying to have some tooling that does a pretty good job for micro-ISVs who just don't want to spend any money. For the Express tooling, one of the biggest shortcomings if you are in that group is you can't use any plug-ins. So ReSharper, as I mentioned earlier, I'm just one of those developers where I really have a hard time working without ReSharper so had I been a victim, if you were, to the Express-only tooling, that would have been just a huge issue for my productivity. But I'm kind of in a weird space because, like you said, I've done some C# development in the past and I spend a lot of time still writing C# and so I actually have a Visual Studio 2010 Ultimate and I actually have a ReSharper license. So for me, to jump in, I have a test runner with ReSharper and it seems to work. My other real gripe with the TDD side was the community. I saw so many blog posts out there saying, "I couldn't figure out this NUnit thing so I just rolled my own XUnit, yada yada,", and I kind of got hung up on those posts for a long time and finally landed on a Silverlight 3compatible version of NUnit and just added that to my test project. The only real pain point at that step was telling Visual Studio, yeah, I know it's a Silverlight assembly, don't worry about it. You know, I'm not deploying this. As far as how those things stack-up compared to the iPhone and the Android side, on the Android side it was awesome because I was using Maven and Maven is kind of like NuGet. If I equate that to the .NET world, it's kind of a dependency management build tool, it's very similar to NuGet. I'd basically said, hey, I want JUnit, and it went out and got JUnit for me. So to write tests, that was pretty simple. Then on the Xcode side, the early Xcode days were actually very painful but I want to give Xcode some credit today. If you're still using Xcode, it actually does kind of what MVC does out of the box when you create a new project that says, hey, I assume you want a unit test project here, right? You can say yes or no.

Scott Hanselman: So it sounds like Test Driven Development is straightforward in all of them. But do you get it out of the box? I mean can you go File New App with Tests and it sets you up? Like does Hello World comes with tests on these three different devices?

Toran Billups: On the iPhone side, if you're using Xcode it will prompt you and say do you want this project set up for testing? So really out of the box that's pretty much the only one that's going to have it all set up before you file a new project.

Scott Hanselman: Is that something that you missed? Like was that a big tax that you paid to go and say I want to find how one tests? Because, you know, it's a good question with each of these communities. Do they value unit testing as a tenet?

Toran Billups: Yes. And of course I came after the Microsoft community there just because my experience on the Microsoft stack, is more of like 80/20. There's certainly a 20% that, you know, they're the hardcore TDD crowd. They're very loud. But there's a good 80% that could care less. I'm not even a purist in that sense. Test-Driven Development is just my style of choice, and for me I just want Microsoft in that community obviously to see the value of the beginner, kind of the Express edition of the micro-ISV. I would like to give a little more support to that person obviously, but like you said it's not that the Eclipse or the IntelliJ experience is much better out of the box, so I probably can't complain too much. I just feel like some of the pain I went through wasn't really helped out by the community so of course I tried to do my part and I actually did kind of a short screencast that just says, look, if you have a File New Project on Windows Phone, here's how you get NUnit up and running . Another issue I had is actually not Microsoft's fault, but it's JetBrains. Their latest ReSharper plug-in actually has a bug in it where when you go to run this Silverlight tests, the test runner just goes gray, like it doesn't know what you're talking about. So I actually did some research there and there's actually a bug with JetBrains so hopefully they fix it. But in the short term, I had to kind of uninstall the ReSharper 6 plug-in and go back to an old I think some version of 5, and 5actually was able to run that.

Scott Hanselman: So I'm hearing a lot of anecdotes from you about this seemed like a good idea. When I plugged that tool into this tool and into this tool, they didn't quite work out. So you're hitting speed bumps that are kind of typical software engineering speed bumps where things didn't quite plug-in in the magical way that one thought they would.

Toran Billups: Yeah, No, that's true, and that's going to be any new platform, right? So you certainly are not going to come out of the box and have an app delivered in two days.

Scott Hanselman: Yeah, definitely. Again, I mean I give these presentations and I give these keynotes and it always looks like you can build an app in five minutes but only if everything works exactly perfectly.

Toran Billups: Yeah, and that's never the case as we know. I just tried to do kind of my part as I was learning the stuff and kind of give back in that way to say, look, here are some serious issues I had and I just kind of took a short screencast to say this is how I solved this so that. . . I feel like at least I'm a very visual learner and I've watched like a lot of screencasts in the last couple of years and I just feel like I get a lot more value from those.

Scott Hanselman: Right. You've got these screencasts of BlueFlix in action and you can kind of see what it looks like on an Android. It looks like a fairly straightforward-looking application. It's got a list and within that list you've got movie posters, information about the movies that you might want to register, and then go pick up at a kiosk location. You do some GPS work. When you're laying out a page like this, did you layout the three different applications on the three different phones to look identical or did you think I want this to look very iPhone-y and I want this to look very Windows Phone-y?

Toran Billups: That's a great question. Initially, when I finished the iPhone app I thought this is perfect. This is exactly what people want. And it kind of mirrored the Redbox experience because Blockbuster Express is kind of a clone of the Redbox. So I kind of had a guide to go from there. When I finally shipped the iPhone app, I thought "You know what? Maybe I'd do the Android app now." I did go in with that mindset that, whoa, I'm shipping the exact same product. It's just on another platform. It must look identical. I mean I need images to be the same, etc. What I found was . . .you know, how many people who are using my iPhone products today also are using the Android product or the Windows Phone product?

Scott Hanselman: Yeah, one. You.

Toran Billups: Exactly. I mean people only have one primary phone. So I got less concerned about that and just kind of went with whatever looks good on the platform. In fact, on the Android side specifically I just completely scrapped using custom PNG images because of the different resolutions. I just had such a poor experience with that that I finally said "You know what? I'll use native controls and just color them red, or blue, or whatever." Because the images, I just wasn't comfortable shipping images to all these different resolutions that, unless you have about 60 phones, you can't really be confident on the Android side. I kind of joke that it's the IE6 of mobile development because that's how it feels when I'm actually working on an UI for Android right now.

Scott Hanselman: Well, how do you lay things out? I know that on Windows Phone 7 you're using XAML which is an XML markup language, and I seem to remember that layout systems are layout systems. I was working at Nike in about '97 or '99 on Java. You know, it was this grid layout and call spans and things like that. I mean layouts are layouts whether or not they're elastic or not, things like that. So XAML and XML, markup is markup. How do you layout an iPhone application? What's that layout look like?

Toran Billups: So the iPhone side was a little different. I initially started doing my Objective-C views in all Objective-C, so I wouldn't even use the layout tooling called Interface Builder. I just kind of said I don't trust -- because it's kind of a drag and drop tool - I said I don't trust drag and drop. That's the devil. I'm going to hand-code this like the real programmer I am. I kind of found that, in addition to being a lot more code, the drag and drop stuff on the front-end for the iPhone really isn't buggy. There are no real issues. I feel like I was jaded. Again, this comes from my unlearning but in my early .NET days I was doing drag and drop web forms sort of development.

Scott Hanselman: Interesting.

Toran Billups: And I remember some of the drag and drop stuff where I kind of bring an AJAX toolkit down and compile and then ship it, and then the next day my co-worker next to me would say, hey, I went to the web page and it says I'm missing this DLL or this component. And I'm like how is that possible? I just dragged that on there and it should have just compiled it. So I kind of was distrusting of drag and drop. But I found the iPhone experience was really nice because, in addition to just dragging stuff on the form, it also gave me a better idea of what it looked like in the preview setting and I never saw, from my limited experience, I never saw a view look differently in the simulator or on a real phone compared to what I thought it was going to look like. So it was very accurate. I just never had any issues.

Scott Hanselman: That's interesting. So Microsoft's failure in the past for some drag and drop things ended up jading you against drag and drop, and the iPhone layout drag and drop works very well.

Toran Billups: Yeah, definitely very well. I honestly don't understand the deep underpinnings of it all, but for someone who wants to ship an app was the view I went in with. It was awesome. I really didn't have any problems with it. The real issues I had as far as layout on any of the platforms was on the Android side. When I was doing my Android development a couple of months ago, the only product that even offered a preview was some version of Eclipse at the time. Of course I was using IntelliJ so I didn't have a single preview. So literally when I was building my app in IntelliJ, I'm just writing XML. I have no idea what this is going to look like. If you remember the painful days of early web development, pre-jQuery and all that good stuff, you'd have a ton of different browsers open. Every time you kind of pushed a new CSS file or whatever, you go refresh them all and make sure it wasn't broken. That's kind of how it felt like doing Android view development except much worse because that preview, you had to actually either, A) load the emulator to go see what it looks like, and you want to talk about slow feedback. The older I get in the software world, the more I'm starting to value fast feedbacks. The projects I've been on that have either failed or not done very well had been because feedbacks are so expensive that what do developers do? They start taking shortcuts. I can't change one variable and check it because it's

too expensive. I'll change two or three and hope that I got everything just right. Well, that obviously eats up more time because if you actually have a bug, you don't know which of those three changes actually introduced it. So that's kind of the experience. In fact when I did the Iowa Code Camp apps recently, I remember putting off the Android app until very, very last because I just had this gut reaction. I was like, oh, I don't want to do that view development. It's so time-consuming.

Scott Hanselman: Okay. So last two questions then. Publishing: was it pretty straightforward to get these three apps into the three stores?

Toran Billups: Yup. Publishing I really beat up on the iPhone experience because as someone new to OS X and obviously someone new to iPhone development, it was the most confusing process to . ., after I had the app built, get it kind of DRM'd and shipped out to Apple because, for one, I was not an OS X user so I really did not understand how certificates and all that stuff worked. But I honestly remember phoning a friend of mine saying, "hey, I'm at the point where I think I'm just not going to sell this iPhone app right now because it's so hard." He's like "I thought learning Objective-C was hard." And really publishing the app was very difficult from that whole certificate standpoint thing. Again I just screencast recently to help other people who are just like 'what in the world do I need to do against this thing in the store.' But in addition to that, of course, the rumors about very tough certifications, they hold true. Apple, I think the first month I tried to get my app in the store, I got rejected six or seven times. It was just a beat down.

Scott Hanselman: Really. For what kind of stuff? Why would you get rejected?

Toran Billups: You know it's probably mostly my fault. One of the very simple ones which, you know, if I maybe had done my research was that when you launch an app and you're in airplane mode so to speak where you don't have any Wi-Fi or you don't have any network connectivity, you can't just let the app kind of sit there and spin so to speak.

Scott Hanselman: Ah, so you need to know you're in airplane mode.

Toran Billups: Yeah. So I obviously didn't know that because it was my first app, my first rejection actually. So I was like, okay, and I eventually worked through all those issues. One thing I will give Apple credit for is they're very consistent so if they go through and find three things this time and I fix those three things, it's done. As long as I do it correctly, it's done and they ship it. One of the pains I had on the Windows Phone side was that they weren't very consistent. Like I would submit it and they' would be like "Great. Good job, Mr. Developer. Here you go." I'd be like, oh, cool. And then the next time I submit like the smallest bug, they find like six things that were wrong with it the first time but I didn't find. So it was very weird because I sometimes, you know, there's no more heartbreak for a developer than "no." So when you're working on this app for X amount of hours and you go to ship it, all you want to see is "Approved, good job, we're going to start making money." So those two, both the Apple side and the Microsoft side, when you get that first rejection. . . it was just very heartbreaking for me as, you know, you can revoke my man card but it was a sad day.

Scott Hanselman: All right. So in closing, let's talk about money. How are we doing? Are we selling these things?

Toran Billups: Yeah. On the sales side, I kind of look at it from two angles: 1), how much money am I actually making, and 2), what's the community or the people, the consumers, like in those ecosystems. As far as raw sales, I've obviously made the most money on the iPhone side, not necessarily to beat up on Android or Windows Phone because I did the iPhone first way back in March so I've had a lot more time to do sales. But on a month-to-month basis, I probably sell, I don't know, maybe four or five times the iPhone apps than I do of Windows Phone. On the Android side, I've been kind of surprised. I haven't even made my money back for kind of the phones I had to buy to do some testing and to actually pay the $25 because what I found about the Android market is, unlike iPhone and Windows Phone where it's like, hey, the app is not even five bucks, it's cheap, that's just how the community feels. I talk to my wife and her friends and other people in the community and it's like, "Hey, it's not five bucks, it can't be that bad." But people on the Android side who traditionally are known for not paying for apps, I found they will pay for apps but they've got to find a lot of value. So for instance, I have usually a spike in my sales when the Android Redbox app is crashing or broken because people just want to reserve a movie. At that point they have no brand loyalty; they're like forget it, I just want to go out and actually rent this movie. I'll pick this up on Blockbuster tonight, so to speak.

Scott Hanselman: Interesting. I heard similar things with the guy that made Angry Birds, that. . . remember they gave Angry Birds away on Android because Android users will never pay for anything. So you're right, there's a definite feel in the different stores. So people can check out your application at blueflixapp.com, and your website is toranbillups.com. I'll put both of those in the show notes, as well as your complete mobile development report card.

Toran Billups: You could probably also include my, I actually have a Redbox app for Windows Phone. It's the only Redbox app there today because the parent company hasn't built an official Redbox app yet..

Scott Hanselman: All right. Cool. We'll put all of that in the show notes, and of course people can read about this on your blog and pick on you in the comments.

Toran Billups: Sounds good.

Scott Hanselman: Thanks a lot. This has been another episode of Hanselminutes. We'll see you again next week.