

An Accidental Programmer's Journey to Offshore Development & the Shortcuts He Learned Along the Way

Webinar recorded with a live audience on Thursday, September 30, 2010

Steve Mezak: So, joining us on the webinar now is Keith Hardwicke. And Keith, we've had a chance to work together for awhile now, but I'm not sure of all your background. Why don't you tell our audience here about your background with programming, and managing programmers, and how you got started?

Keith Hardwicke: Well, I got my PhD at University of Texas in Electrical Engineering and Control Theory, and started working in a research lab, called Applied Research Lab in Austin, Texas. And like every research lab that I had been to before, I got to the research lab and all the physicists immediately asked if I programmed. And so they said, "Here you go." So there I was programming again, even after I got my PhD in electrical engineering.

But it was really interesting and exciting to build systems that had never been built before. So I decided that it was a good thing for me to learn. And then I had the opportunity to move out to Silicon Valley to make my riches.

[laughter]

Keith: You know the usual, "Oh, I'll start a start up company and I will become famous and rich." So I came and I helped start a start up company. And lo and behold, I made an OK living, but I did not become rich. [laughter]

Keith: over the years I learned about all the IT infrastructure that's involved with delivering software solutions to the general public. And I also was introduced during this time period to the concept of outsourcing, to using global resources, rather than just resources in the United States to achieve goals that probably could not be accomplished with just resources from one country.

Steve: So you're doing K&L Consulting right now. How would you describe the goal of your business, or the goal of the services that you're providing?

Keith: What we're providing is a virtual software development shop that's available to other companies, but mostly companies that do not have formal expertise with software development.

Companies that, for some reason, have a need to do software development, but don't have a software development practice, that's the target, the sweet spot, of our audience. Because we can come in and help them with all parts of software development life cycle.

Steve: So then, the clients come to you with some kind of idea for a software application? Is that usually the starting point?

Keith: Well they either have an idea for a software application or they have a need, like they need to implement a CRM system to make their sales force more effective. And even implementing software, today's enterprise class software, takes a certain amount of IT expertise that the normal company does not have.

Steve: So back in the beginning of the year was when we first met. And you had a client that started at that time with an interesting application of doing some sort of video thing over the Internet.

Steve: Why don't you tell us a little bit about that?

Keith: Well, this was a really interesting client. The client actually is in the manufacturing space, the semiconductor equipment manufacturing space. And has never done software in his life. He posted a job listing on monster.com, or CareerBuilder, I can't remember which. He said he got on the plane and went to Chicago, got off the plane and there were like 400 applications.

Steve: [laughs]

Keith: And he didn't really know where to start so he just started with the first one, and then when he was on the fifth one, he was like, "I just can't do this. It is just too much." And he's like, "There's got to be a better way to do this." And then he thought, "Gosh, if I could just see some of these people, it would be much easier." And then I got a phone call the next week. And he was like, "Can we like do a video interviewing software?" And then I said, "Yeah, in concept you can do that.

So we've been building that and it's been a real adventure

Keith: So we had this need from a person who'd never done any software before. His only requirement was he wanted it to be something that he could use.

Steve: Yeah.

Keith: So we put together a pilot and basically, the way that K&L likes to do things is we like to put together a set of artifacts. We try to produce the minimum set of artifacts we feel is necessary to get what we want accomplished. But we do feel like we need something at the beginning of the sprint that we can say, "Do this," and that will start dialogue.

And we gave a bunch of parameters for the non functional requirements. We wanted to have automated testing of certain varieties. We wanted to make sure that they used the main models.

Steve: Right.

Keith: I contracted with an Indian firm. It seemed like a good opportunity because they were relatively inexpensive and they claimed that they knew Ruby on Rails and that it was one of their major languages.

Steve: OK.

Keith: And so we finished the pilot. This was like in December, I think, of last year. When they were done we took a look at the code. Then within about three minutes, I threw up my hands. Based on what I had seen they missed the whole point of Ruby on Rails. At that point, I was up on Google just like anybody doing random searches on Ruby on Rails outsourcing. And then I took the top 20 and I just sent off emails and called. And I got some replies.

I was kind of surprised. Things just didn't go to the ether. I got a call back from a guy called Steve Mezak...

Steve: [laughs] That's right.

Keith: . And then I got a call back from somebody in Boston. And Steve listened to what I had to say and looked at my documents and immediately thought of Svitla. And the person in Boston, his firm of choice for this particular application was a firm in Argentina. So at this point I thought, the more the merrier. I am not just going to have one this time., so I sent it off to both teams.

Steve: Let me stop you and ask for the documentation you were providing, and you called it 'artifacts' earlier. What were you providing to the Indian company and the Argentinean company into our partner in Ukraine?

Keith: Well, when we do our artifacts, we try to give some sense of what we call a domain model. By domain model, the developers get the idea of what the database scheme is supposed to be like. And they also get the idea of what sort of objects are really present in the system.. And then we overlay with these Visio documents. We are trying other tools as well, but Visio documents we call wireframes. And then the wireframes, each one of the interactive items has a call out.

And then that goes into what we call functional spec that actually says in English , when you click something or you hover over something, what exactly happens.

Steve: Right, right. So in the case of the wireframes, you are giving the developer the idea of what the user interface should look like and how it behaves. And then the domain model explains the interrelationship between the kind of entities that would be represented by the software. So in the case of interviews, you are going to have a video, what other kinds of things are in your domain model?

Keith: Well, we have things called questions and stats. And then we have all the different types of users and all the organizations you would expect in an HR, Human Resources setting. Then we even have something called the job board or career center. So we basically try to take the real world and as closely match as it is feasible. And then that domain model actually, we derive a glossary from that. When we write the functional spec we write it kind of to exercise the domain model. So when you are reading the functional spec, you'll see all sorts of domain model terms in the functional spec.

Steve: Right.

Keith: Basically we are trying to tie the user interface to what we call controllers that exercise the domain model.

Steve: This sounds like quite a bit of information that you're providing the developers. How many pages would you say it ran to?

Keith: Well, a typical wireframe set is about 10 pages. It's each one of the screens with each button on it and each scroll down. Then the functional spec basically explains the interactivity of each one of those calls, so it can be about 15 pages. Then the domain model is like two diagrams. It's not huge. We don't always have to do all of those artifacts. After a while, the domain model is the same. You're putting new user interface views on top of the same domain.

The alternative is to not do that, and then basically your offshore team is either coding directly to the database and they're choosing their own schema. Sometimes that's not what you want.

Steve: I think it's a good balance of information. It's 25 to 40 pages of information. This is for your pilot, and perhaps for each sprint along the way, that's the kind of level of information that you feel you need to provide in order to guide the team to what you really want.

Keith: Correct.

Steve: Let's see, we were talking about the experience that you've had at your other companies and how that prepared you for the consulting business that you're in now. At the companies you've worked at, you've had an opportunity to do outsourcing, then. Why don't you explain that a bit to us?

Keith: At first I treated outsourcing pretty lightly, because I really wanted to program and do product development. I thought outsourcing would be good to do for the quality assurance and the test function, . But then we got a new CTO, and he decided that it was time to engage in setting up a real off shoring effort on the order of having four simultaneous Scrum teams working in India.

He tasked the onshore team with communicating what we wanted. Well, we had spent our whole lives building, and suddenly we were told, "Nope, you're not building anymore. You're communicating what you want built."

over the next few years, a lot of frustration, a lot of time spent trying to figure out how can you communicate what you want without actually programming it? How could you do it in short period of time so the total process is not more expensive than just hiring a couple of smart people in the United States? How can you actually reap the benefits of outsourcing?

So I struggled., and I came to the conclusion that we would put a lot of emphasis on a lot of artifacts, and really all they would use is the UI, the wireframe.

The rest of the stuff they would just put in the trash or some closet somewhere. They'd get the wireframe out, and you'd get back something that looks just like the wireframe, and you had no idea what it was going to do.

[laughter]

Keith: So then it was like, OK, well, at least they're using the wireframe. Then we were like, well, is there any way we can tell them what to do when they click on the buttons in a very succinct way, in a way that's unambiguous? That's when we started building these small functional specification snippets. Basically it says, "This is the element. This is what it looks like. This is when it's active. This is when it's inactive. This is what it does when you hover over it. This is what it does when you click."

Steve: Right, like a button or a list or something like that on the screen.

Keith: Yeah.. Then we started passing those across, and that worked pretty well.. Then we noticed that the database schemas that we got back were just horrendous, and we were like, we need to have some guidance on the sorts of duplication and stuff they're putting into the database.

So then we got the idea, we'll put a small, thin domain model on there, and we'll tie that back to the user interface spec. Well, that seemed to be the ticket. It had to be tied to the UI eventually. If it wasn't tied to the UI in some way, it wasn't going to get programmed. It was just going to get people to look at it and say, "Oh, that's nice," and put it away.

Steve: What I've seen, too, is if you go down into much detail of the database schema and do an entity relationship diagram or even give them the table definitions, then your user interface suffers because it looks too much like the database tables and not the way it would normally be used or not the way it would normally fit the model that a user would have in mind on what the software is supposed to do. Did you run into that problem also?

Keith: We learned pretty quickly that if we started specifying how, "This is how you do the schema," then they'll do exactly what you say. Then you're like, [laughter]

Keith: I learned pretty quickly that if it's going to be an effective thing, they have to be free to make their own choices, but you have to guide their decisions with reasonable artifacts, and artifacts that are somehow tied to the UI. If they're not tied to the UI, probably they're not going to get used. So over the years I learned the artifacts that make the most sense, and we zeroed in on those. The other ones, all the sequence diagrams and things like that, trying to explain exactly what happens next in gory detail, that stuff...

I read books, and they all say to do it, and I've just never been able to really make that work in an off shoring situation. You spend so much time doing the English for that, all the wording. I just haven't been able to go to that level of detail yet.

Steve: That's interesting, because another thing that I've seen that can work well but also takes a significant amount of time is to create some sort of storyboard or a clickable prototype. I've done it with HTML where you have the HTML page, which you're representing as a wireframe in your artifacts. But let's say you have an HTML page which is pretty much like the one you want to have in your web application. Then you create another page, and the button on page one clicks and brings you to page two. You can get a feel for not only what the page looks like, but also the sequence in which the pages are used to carry out some particular function. Have you ever had...

Keith: I've never had success with those sorts of... What did you call that, a functional prototype?

Steve: Applicable A clickable prototype.

Keith: A clickable prototype. The reason is the people that do the applicable prototype are the same people you would want actually building your application. The people that I want doing my wireframes are people much closer to the customer and their needs.. I don't care if they know any HTML code at all. Matter of fact, that might even be distracting for them.

It's just been my philosophy that the moment you start going away from just doing a flat wireframe with explanation, and I know different people have different abilities to imagine.

Steve: So then when you started K & L Consulting, it's really to use these kinds of techniques that you learned through your own sometimes painful, sometimes happy experiences in making this work. Then this client came along, as we were talking about with the video software, and then you were in a perfect position to then go for their idea, to the wireframes, and to the other kinds of artifacts that you need, the domain model, and provide a good description of what needed to be done.

Keith: There are all sorts of companies out there that are getting drug kicking and screaming into the software arena. They don't have the capability to build a software, and they can't afford to make the investment that it really takes to do it right.

So I thought, "Wow, what if we set up a business that had sort of a virtual software development organization and was able to meet the needs of this paradox," that there's all these companies that need to do software development, but there are so few that can.

Steve: Yeah, yep. So you have a really great handle on what's required at the front end. Then at the back end, it's programmers and outsourced teams wherever they happen to be in the world. So it's the perfect fit for what Accelerance does because we have those teams. So when you came to Accelerance, you had this pilot project all defined out. And we selected one of our partners in Ukraine. So maybe you could explain a little bit about how that worked out, that initial pilot.

Keith: OK, well, the pilot, as I said, was between an Argentinean company and this company in the Ukraine called Svitla. I passed out the pilot to both vendors and they got started.

. I interacted with the programmer that was put on both sites. What I learned pretty quickly was two things: that the programmer that had been put on the project in the Ukraine spoke better English than the one in Argentina. And the other thing I learned was that the programmer in the Ukraine seemed quite a bit more familiar with software architecture concepts than the programmer in Argentina.

Ultimately the reasons I went with Svitla, One was the English was better, and that made a difference. Two was the programmer at Svitla asked very intelligent questions and communicated better than the programmer in Argentina.

A certain one of the Rails gurus checked the code, make sure it was OK. I think that the Svitla was slightly better. I think Svitla took it a little more seriously than the Argentinean company, as well.

I made my decision, then we started our first sprint.

Steve: OK. you've mentioned Ruby on Rails was the technology. How did you make that choice for this particular project? Was that your choice or was that dictated by the client?

Keith: That was actually my choice. The reason I chose Ruby on Rails is that I've learned. [laughs] I've developed a lot of software, and. Normally what happens is if they're given three months to do a project, they'll fight for the first two months over the architecture. And then the last month, they pull all nighters to do the project.

I started reading about Rails and it's like, "We don't want it to be everything to everybody."

And I thought that was pretty cool. I found out that the many domain models I was doing matched pretty close to the models in Rails.

Steve: the other good thing about Ruby, too, is that it fits into the agile kind of development process you are adopting of short sprints.

Keith: Yeah and it also has a strong philosophy automated testing that I really appreciate quite a bit. Some people call it test driven development. Some people call it behavior driven development. But it gets your quality assurance and your test people who are involved in the process up front.

We are experimenting with ways of having the test organization write what we call cucumber scripts in English that actually will drive the application to the various tests that we wanted them to go through.

Steve: Yeah, yeah. That's impressive.

Keith: That's very, very exciting. We set up a continuous integration server that allows us to Do the build and then if it fails, we don't promote the release. So we have a fairly

sophisticated deployment process. And this was all set up by the team there at Svitla. Very, very sophisticated IT software professionals that I've come to expect a lot from.

Steve: did you specify all that for them to then go and do?

Keith: Nope. I did state my wishes. I said, "In the best of all worlds, we would have a continuous integration server and we would have automated integration testing, that if failed the automated integration test, then it wouldn't promote." And that's what I said. And sometimes you have to watch out what you say because then they are like a bulldog and they want to go do that. [laughter]

Keith: And you are like "But wait a minute, I want you to do this functionality." "No, I'm working on this other thing." OK. You got to pick your battles. So in this case it is very important and with this sort of agile mindset, if you do agile programming and you don't have the protection of automated testing, well, soon you are going to get in what I call Fragile Land.

Where you pick something in one place, you deploy it and you broke five other things. You have got to have some protection and so it makes agile work. The whole behavior driven development model.

Steve: Yeah. So, how did Svitla compare to other vendors? Ukraine is one of those countries that I don't know, some Americans may not have heard of. It is in Eastern Europe. Culturally did you find any challenges, any issues?

Keith: I have done quite a bit of outsourcing to Russia before. I found the people in Ukraine to very much have the same mindset as the Russian people. They seemed very smart. You would give them hard problems, they liked to attack hard problems.

Steve: you didn't visit before making the decision to hire them. You did the pilot project for that, but then subsequently you have the chance to visit?

Keith: Yes, I did. I decided to go out to Svitla because I needed to understand what I could do to streamline the process. In order to do that I need to meet people face to face. One of Svitla philosophies is to put what they called a lion developer on top of the project and then have a team of I don't know what lions are called, not herds. A pride?

Steve: A pride, yeah, pride of lions.

Keith: So it is more like a pride of lions with the head lion leading the pack... And so, I had my lion developer and I needed to talk to him face to face about certain things It turns out that some of the ways we were specifying that they do things were not optimal. But they were challenging and they enjoyed doing them. So they did them anyway even though it took them a week longer because they enjoyed the challenge. And I thought, "No, that is not the point."

[laughs] So I was able to communicate that was not the point. We were not trying to provide them puzzles for them to solve. We were trying to produce software.

So now they push back. I have them pushing back when we produce something, some user interface that is not as easy to implement as maybe perhaps an alternative interface that would get the job done.

Steve: What would be a good example?

Keith: We had a software component that we called job board admin. We specified it, similar to the way I think you would do a desktop application, with one page and a lot of tabs.

Steve: OK.

Keith: And it turns out that had we done it as a normal web application with many pages, it would have just made the process maybe a week and a half faster out of four weeks. But they were enjoying trying to figure out how to make this work. [laughter]

Keith: So that was one thing I learned about the culture there. They enjoy challenges,. You know, you say, "I want you to do this. If you think of a better way, let me know and then we can do that." And, then I made it part of their job. It's your responsibility to tell me when we're making it hard for you to do stuff. And they've really transformed since then, much more collaborative environments. So I'm very, very pleased with that. I'm pleased that I got to go out there.

Steve: Yeah, I'd say the one thing that I've heard about Russian and Eastern European engineers, and Ukrainian engineers in particular, is they're very direct. There's no beating around the bush. They will tell you if you're wrong even if you don't want to hear it at times. [laughs]. Did you find any extreme cases of that or concerns about that kind of pushback?

Keith: No. I'm used to, because I've dealt with Russia and the Ukraine both, usually they're not very shy. In this particular case, I learned that sometimes they look at it and they go, "Wow, that would be kind of fun to try. Let's go ahead and do it." But that isn't what I wanted. I mean, we're talking about very smart people with Master's degrees in math that have a creative mind and want to do new things., it's definitely not a sweatshop. It's definitely a team of professionals that are highly educated and want to do a good job, very impressive environment.

Steve: So it sounds like you started with an interesting project and a need, and without really targeting a specific country, you found Svitla in Ukraine to really be a good choice for you. What were your expectations going into it? Did you have any places in mind you definitely wanted to outsource or wanted to avoid?

Keith: No, I was really open to just about anything. I really wanted an organization. My criteria was it had to cost less than the United States, and that was pretty easy to meet that expectation.. Then the other expectation that I had was that the organization needed to have a deep understanding of Ruby on Rails. Because I didn't want an organization that had a bunch of PHP programmers pretending to be Ruby on Rails programmers.

What I got was something I had never imagined. I have not lost a single developer on this project, and it's been going on for like seven months. That's new to me. Normally, it's like, "Oh, Joe's not here. You get Frederick." [laughs]

So Svitla has what I consider to be a very good characteristic of longevity of their people it seems. And I don't have to worry about.... Walking in one day and seeing that my lead developer is gone on a different project or has left the company.

So it seems like they have a certain stability I wasn't anticipating. They actually, most of the people on the project are all based in one room in Kiev, and I think that really helps out a lot to have all the developers and the test person in one room.

And the amount of expertise, IT expertise that I've gotten, I'm quite surprised, actually, Normally, with some other outsources that I've done, I've ended up doing all the IT and they've done the programming.

Steve: Right. Where do you draw the line between programming and IT?

Keith: Normally, if it has to do with the functionality, I call that programming., mostly the IT stuff is like automated deployment processes, all the monitoring, the alerting when things go down. All that I consider to be IT. And we do fire drills. "Take this server down." Does it affect anybody? [laughs]

Steve: Yeah.

Keith: . We'd like to have a system where any one component would fail and the whole thing would still be up.

Steve: Yeah, the IT stuff is what sometimes I call deployment. So it's one thing to create the application, but it's another to get it deployed out there, and up and running, and staying up and running, and to fix problems like this along the way. It seems like they are great at innovating in those areas as well.

Keith: Yeah. That was one thing that surprised me, because I was expecting to do a lot of deployment work myself. We have to use Amazon EC2. We have to make it to where it has certain reliability. You have to be able to lose a frontend server and it still has to function. We need a load balancer. I just told them what all we needed and then we just started after it. We had these Flash media servers that are called Wowza servers, and we rent those from Amazon as well. They learned to control those. It was basically IT, and they learned to map the videos directly onto the S3 using something called S3 Fuse. They did all that without me even knowing it.

Steve: All right. Good. Well, any final things you want to say to wrap up or to let our audience know about outsourcing to Ukraine?

Keith: Well, I didn't talk as much about the interaction I had with Accelerance. And I wanted to say some things about that if that's OK.

Steve: Sure. Yeah.

Keith: So what Accelerance provides me, because I am a solution provider to various companies, is I can walk into a company and they can throw the most abnormal technology at me. They say, "Oh, we want Joomla". Well, OK! So then the first thing I normally do is I come back and I call up Steve. I say, "Hey, you know, anybody that does this." . And Accelerance is going to have a preferred provider that does that, and it will do it with an acceptable level of confidence. So I need more than just Svitla. I also need Accelerance to do what I do.

Steve: Yeah. Well, thanks. And that's our goal, And the goal of this webinar is not to convince everybody that listens that you must outsource to Ukraine; it's the best place or the only place on the planet. It's not, but it's a great choice for certain types of skills that are needed. And, you know, there are tradeoffs. So we are really happy that this worked out well for you and it works out for a lot of other clients as well.

All right. OK, good. Well, stick around. I'm sure we're going to have a lot of great questions from our webinar attendees.