

IT Matters Podcast - Episode 42

📅 Fri, Oct 09, 2026 3:32PM ⌚ 31:34

SUMMARY KEYWORDS

IT leadership, technology evolution, automation, e-commerce, troubleshooting, command line, GUI abstraction, technical skills, curiosity, problem solving, continuous learning, digital privacy, remote work, collaboration, technology strategy

SPEAKERS

Neil Stein, Keith Hawkey, Aaron Bock



Aaron Bock 00:00

Welcome to the IT Matters podcast, hosted by Opkalla. We're an IT advisory firm that makes technology easy for your business. Our vendor-neutral technology advisors work directly with your team to assess technology needs and procure the best IT solutions for your organization. On this podcast, expect high-level expertise from our hosts, plus experience-driven perspective from the leading experts on topics like AI, cybersecurity, industry-focused IT solutions, strategy, and more. Now, let's get into today's discussion on what matters in IT.

K

Keith Hawkey 00:35

And welcome back to the IT Matters podcast, hosted by Opkalla. At Opkalla, we help IT teams understand the busy marketplace of technology strategy and services with a data-driven approach. On this podcast, we invite technology leaders to discuss the challenges facing the modern IT department. My name is Keith Hawkey, Technology Advisor at Opkalla, and today I'm joined by Neil Stein, Senior Vice President of Technology Services at OrthoCarolina, whose path in IT didn't start exactly how you'd expect. He actually found his first love with chemistry while going through school, but Neil quickly realized that chemistry was interesting, but technology gave him something even better—an endless supply of problems that somehow become your responsibility, whether you created them or not. He's been solving those problems ever since. Neil came up in a generation of IT where there wasn't always a nice GUI or SaaS platform or AI assistant waiting to solve problems for you. Sometimes you had a command line, limited documentation, and the confidence to type something in and hope you didn't take down production. And over his career, Neil helped build e-commerce in the early days of the internet. He's led infrastructure and technology operations and has taken on an increasingly complex leadership roles, all while maintaining a hands-on, figure-it-out mentality. And that's really what we're going to dig into today. As technology becomes easier to consume, are we actually becoming better technologists, or are we losing some of that curiosity and problem-solving skills that created great IT leaders in the first place? So, we're going to talk about that evolution, Neil's career, and some of the lessons he's learned along the way. Neil, welcome to the IT Matters podcast.

N

Neil Stein 02:40

Thank you, Keith. Great seeing you today.

K

Keith Hawkey 02:43

Good seeing you as well. But per tradition, before we start the conversation, we're going to start with two truths and a lie that is IT technology related. So this is a little bit of a surprise, but are you? Do you know how to play two truths and a lie?

N

Neil Stein 03:02

I do.

K

Keith Hawkey 03:03

Right. Pretty simple. I'll state three. I'll make three statements. Two of them are true. You're going to try to spot the lie. Are you ready?

N Neil Stein 03:11
Yes, sir.

K Keith Hawkey 03:13
So first, the first computer mouse was made out of wood. the second?

N Neil Stein 03:23
That's true.

K Keith Hawkey 03:25
Okay, we're going bull. He he knows without even finishing the statements here. All right. So second, that was all good. The first ever online purchase was a pizza ordered through Pizza Hut. Third, the term debugging became associated with computers after engineers found an actual moth inside of their computer.

N Neil Stein 03:56
Okay,

K Keith Hawkey 03:58
which ones of those are do you think are true, and which ones

N Neil Stein 04:01
stick with the first one being true as a as a wood mouse. Second one I'm going to say is false for the ordering of the pizza. And interesting enough, I just was reading a book the other day about the bug and the moth, so I'm going to say that one's a true statement.

K

Keith Hawkey 04:19

Well, there you go. That's that's pretty sharp, Neil. You are correct. The first mouse computer mouse prototype was built by Doug Engelbart's team in the 1960s, and it had a wood casing. And secondarily, in September of 1947, engineers working on the Harvard Mark II computer were troubleshooting a malfunction when they discovered an actual moth trapped in one of the machine's electromechanical relays. They removed it and taped it into their logbook and wrote. From now on, solving computer problems is going to be called debugging. So, believe it or not, it actually came from a bug. And Pizza Hut did launch an early online ordering experiment in 1994, but it wasn't technically the first online purchase. There were earlier transactions that predated it. A lot of them, you know, were commercial transactions, you know, online orders. So they were close, but not quite the earliest that was recorded.

N

Neil Stein 05:35

Okay, that's good to know.

K

Keith Hawkey 05:39

Well, let's dig into it, Neil. Can you tell us? You studied chemistry in college before finding a love for technology. What happened early in your career that made you realize technology was where you were going to have a bigger impact on the organizations you work for?

N

Neil Stein 05:58

Yeah, so I enjoy chemistry. Still do. I don't do a whole lot of it in my current career, but back in the day, there was a lot of math and a lot of work needed to be done after your experiments, and that always needed to be double checked or sometimes triple checked by another individual. And so, some of the things I did early on in my career was creating simple software programs to actually go do that work, so we can minimize the amount of impact of individuals needing to actually go do that work. It did speed up the overall process, and in that time, it it was a challenge. I do enjoy puzzles, and so how do you tie all those pieces together, making sure that the math that's being done in your program is accurate, right? The number of decimals, everything needs to be taken into account. So rounding was not the same, you know, depending on which calculator you might have used compared to what actually software was being used and what hardware that software was running on. So it was an interesting balance of understanding how do we get our results quicker, and how do we drive some? And I'll call it back then. It wasn't known as automation, but how do you drive automation of getting the data out?

K

Keith Hawkey 07:10

And you were building automation and working with things like e-commerce early in your career, and in the early 2000s, before those technologies were nearly as accessible as they are today. What are some of the challenges of spinning up an e-commerce infrastructure in those early 2000s? How is that different than? I mean, I imagine today it's actually fairly fairly simple, but I could be wildly incorrect.

N

Neil Stein 07:38

No, well, some of it was just the newness of everything. We take the ambiguity of SSL or HTTPS as an easy one today to get a certificate. Back then, it wasn't as easy. There were patents you need to manage, and you had to use special software just to have it encrypted traffic time to to do things. Right. So today, we think just like that, things are encrypted and decrypted. Back then, it added time. We're delays. Our goal was 10 to 15 seconds for page loads. That's an incredible. Just imagine going to an e-commerce site today and you wait 10 to 15 seconds for a page to load. You're not going to go do that. Then storing the data, making sure you had good transactions as you manage the inventory, most of those things back then were manual process or in spreadsheets, but in separate systems. Pulling all the pieces together was another type of puzzle. Understanding when things rotated, time zones weren't as easy to be managed today, and so what were you centralizing your business on? Which time zone would you operate all your systems on? Nowadays, it's just something we do naturally. We use a UTC or we use a local time zone. Back then, it wasn't the same. When did you do maintenance? Right. So Monday through Fridays, when systems were available, evenings and weekends, most systems were offline because that's when businesses closed for the weekends. Now you're operating in a different paradigm. Who's supporting the systems? So there was a lot of differences back then. There was no go buy something. There was no off the shelf. It was you had to code it. You needed to know how all that code worked in case it was an issue. How'd you manage where things were going, and so that was the hurdles in the early day of creating an e-commerce site.

K

Keith Hawkey 09:28

Yeah, I remember. I remember watching a video a few years ago about how complicated time zones and programming programming different time zones and how that correlates to transactions from the ground up can be, and I mean the the end result of the of this advice was to go find someone that's done it before, thank them for all of their work, and just just just just take their programming and and add it to whatever you're you're looking to do. Because especially from a global perspective, time zones can be increasingly complex. I don't know if you experienced any of that.

N

Neil Stein 10:10

I have. So back in early days, believe it or not, we solidified on a unique time. It was 1130 Eastern time. Was when our days rotated over, and that allowed us a little bit of management around things. It also pulled it out of what time zone we're doing for our days. In itself, from my side, it worked beautifully. But then, when I was working with the marketeers, they were trying to track data, and now the data was skewed by 30 minutes each day. It was pushed to the other day. Over time, it caught up, but it did create unique problems. And as you pointed out, Keith, when we start looking at global, how do you deal with before and after when your systems are running? And then when we went down for maintenance, how did you handle those type of things? And even broader, we built the DMZ. The DMZ had your database, everything in it, because you needed to have a very controlled environment to make sure that your networking equipment didn't go offline for maintenance between your database and your app server. And so, how did you manage all those pieces when it came to just timing? And then, lastly, yeah, not too long ago, 25 years ago, people work nine to five, and if there was a production issue, they might have caught it up back on Monday morning. Well, you can't do that, so time zones need to take that into account. So you start looking at where your team is working. Are they working just on the eastern side of the U.S. Are they spread across all four time zones. Do we also go and look at if we're an international company? Can we have some sysadmins or engineers in other parts of the globe? So some of the things I did early on is how do we get team members that aren't local getting access locally to do your work? Last piece, Keith, is back in the early days, network speeds weren't what they are today, so I remember typing in on a command line, and you wait two, three seconds, and the characters come up. And so when you type, you get really good at touch typing, and you don't make mistakes because then you would have to backspace maybe 30 characters. And so as you go forward, just those simple things added up to the complexity of really needing to know how the systems work, how can you get through a problem, and how do you work with people that there is no Zoom, there is no Teams, there is no Google Meet. Nothing was really in existence. You had Webex calling and maybe some Skype. That was it. So how did you deal with people that weren't in your time zone easily in your company,

K

Keith Hawkey 12:43

Neil, you you bring up a great point, and your generation of IT professionals had to understand what's happening under the hood with that command line scripting infrastructure troubleshooting, where someone entering IT today can accomplish a tremendous amount through a GUI. Do you think that abstraction is changing how younger IT practitioners learn to troubleshoot and solve problems? Could that mean you know what could that mean for the next generation of technology leaders?

N

Neil Stein 13:22

It definitely is changing. So, Keith, if you only spend it out to a little bit more broader around call and technical assistance centers, TACK, there was not a whole lot of TACK that you called back then. So, you had to reach out to your peers, even visually seeing things occur. So, yes, the GUI is probably calling a command on the backend, but a lot of people don't know what that command is. What are all the parameters around? And then GUIs change frequently. Command line. Then you had your historical command line, and you added more. And so I think it is creating problematic pieces because you're separated from that system. A lot of times, you go into a command console that talks to your environment. It's not on that environment. Is it a bad thing? I don't believe so. But I believe that the things that's missing is truly understanding how a system works, how a firewall actually creates an IPsec tunnel, what is needed, how do a load balancer really balance what's actually coming through the system. How do you use virtual items such as IPs and everything else? And so yeah, the GUI allows you to quickly go through it. On top of that, you have a lot of templates. So what I see a lot of engineers nowadays are using a template that was created by somebody else. And now when I go ask them, hey, listen, we need to do these features. They're like, I don't know how to change the template. I got to go figure that out. And so just the basic understanding of how things work. Lastly, it's just pulling things apart. So yeah, I remember the day where you brought up different hardware. You had your big giant test lab, test labs. Going away because a lot of things are software as a service that breaks things as well because you're going out to a vendor's GUI center that talks back to your hardware. So the separation and extraction and extrapolation is getting broader. It does create a reliance on that vendor. Does the vendor like that? Yeah, that's that holding. That's that glue to stay on that particular vendor. But the command line, understanding how things operate, allows you to create and do very magical things by creating your own GUI on top of that yourself. So it gives a layered approach of how much do you dig in, how much do you not. Last piece, Keith, is the GUI does give an opportunity to lower costs, so you can bring technical resources in and more junior to do more advanced things. But that has its hidden costs because when something truly breaks, you don't have that knowledge. Sometimes mid to senior level people are really required to troubleshoot the problems, and what I always say is, IT is not here for when it's beautiful and sunny and it's a wonderful day. The IT needs to be there when there's problems, and you really need to know how things work, especially in 2026, where five minutes can seem like an eternity. When when I started, people were raving. Okay, it's for five hours. We'll deal with it. Nowadays, you can't be down at all, and so knowing how the systems work is very important.

K

Keith Hawkey 16:28

That brings me to brings you to another good point of of these younger technologists, younger IT leaders that are existing in this gooey landscape, they're going to become the mid and senior level technology leaders of the future. What kind of advice would you give a fresh computer science graduate getting into the space, eager to learn, yet the environment in which they are going to spend their first cycles as an IT practitioner are very different than your first cycles, and where the things that you had to learn, how can they? Because because your your your analogy around Understanding the templates and how they work is is resonating to me because how do they become that more mature technology leader when they don't have those fundamentals? Like, what what type of advice would you give a younger IT practitioner?

N

Neil Stein 17:37

First and foremost, never stop reading. Read everything you can get your hands on. Every article that's out on the internet, every magazine, every book. Listen to a lot of people who've done this work. Dig in, and then on top of that, play. Right. So even if you're starting out with just a simple Raspberry Pi, go do those type of things. Have VMs running on your local computer. Don't get too stagnated that I'm a Windows or Mac or Linux type of individual. Work on different systems so you keep that perception and perspective that's different. And then the last piece is ask a lot of questions. Online forums. Ask how they're doing things. This is what I'm doing. Talk to your peers. Question things. Be curious. Those are the fundamental pieces as you're a leader. As you go up the ladder, you can't stop doing those type of things. You need to be hands on to some aspect, and it could be in your hobbies. But just be curious of how tech works. Don't just assume, hey, I plug this in and away we go. Right. Just understanding how everything is operating, you know. There was questions about going from IPv4 to V6, right? V4 is definitely easier to read for a human. V6, yeah, technically, but then you have a lot of more rules you need to understand. So understand those pieces. What does it matter? And then talk to junior people, be that leader, right? And so, one of the things that I see, you brought about GUIs in several areas, things are made easier, but we're losing that apprentice type of thing. And so, in the era when I was growing up, there was always more senior level people that I would have those conversations with. So, user groups in your community definitely be part of those. Be either that master or the apprentice, and it's never changing, right? So even younger people than me, I learn from them. So always be once again back to that curious. Never say, "Hey, I've been doing this for 30 years, so I know everything. You don't. Things change, and so I think, Keith, that's the driving force. Yes, I do a lot of paperwork. I do a lot of budgets nowadays, but I'm still curious how things work. I like to understand how they how they fall together. How do they fall apart? Last thing that I give advice is couple couple words. I don't understand. Can you rephrase that? I didn't understand why you explained it to me. Be that person. Ask those questions. Say, "Hey, listen, I have no clue what you're talking about. I think Steve Jobs mentioned in one of his interviews, "I don't have the foggiest idea how that works. That's a great statement. Be that person to ask those questions of of everybody because it gets everybody in that room thinking of I don't know I don't know how this really works. So if it breaks, what am I going to do? Because it never breaks when you have a lot of time. It breaks at 455 on a Friday when you have busy weekend and you need to go fix it. So that's my thought, Keith, of constantly working with team members.

K

Keith Hawkey 20:44

I love being around the people that can formulate the best questions. They are always far more compelling and interesting than those that seem to think they have the right answers all the time. I mean, nothing is sweeter than just a well-formulated question that provides insight in the question itself of of how it's formed. I mean, that's you know, at one point does a technology leader need to stop being the person with the best answer and start being the person that can ask the best questions.

N

Neil Stein 21:20

First, I keep going from chemistry, which is always a research type of approach. Right, so I work in pharmaceutical research. I think from day one, as soon as you graduate high school, be the person asking the questions and never stop. You will endeavor to be sometimes the smartest person in the room, but you also be the person who doesn't know enough in that room, and so I don't think it's a point in your career. I think it's throughout your entire career from day one, and if you can start before you graduate high school of asking all those questions, I think that'll make the best leader. I think that'll make the best technologist for the next 20, 3040, years going down the road. How can we make this nuance better? The approach is my goal: 1% better every day. It's not much to ask, right? So, can I learn something differently? Even this podcast, I'm a deeply introverted person, right? So, if I don't speak, I'm good. Yeah, but I know I need to be out there. I need to go do those things. And so, Keith, I think constantly asking questions, understanding. Hey, yep, Friday's my paperwork day, but give me a reason that I have to be pulled into the lab. If there's a problem at two in the morning, call me. I want to be on that call. I want to hear how people work through things. I want to look at the aviation industry to see how they work through issues. I want to see how the military works. I want to see how civilian works. I want to see how users use my tools. Ask those questions from day one. Why? Be a little kid. They always ask why, right? Why? Why? I think that's the the time, Keith. Whenever you're ready, start asking those questions.

K

Keith Hawkey 22:58

Yeah, those are those are really good points, Neil. Technology is often is often associated with progress, but there are times when choosing not to adopt a new technology is actually sometimes the more innovative decision. Are there times in your career where that you had to say no, that you felt that that was the best course of action for the organization, for your team, or at least adopting a technology in a different way.

N

Neil Stein 23:29

Yeah, there's always that question, right? I love technology, but there's certain things that I don't have in my home because I don't think they make sense, right? And so I don't need a thermostat changing the temperature 30 times a day, depending on who's there and who's not there. I just don't see it. But I do like automation around certain lighting that goes on at sunset, and that constantly is changing, and other pieces. So in business, it's the same philosophy. To have it, yeah, we need Wi-Fi. We need all those type of things. Tracking where employees are in the building, maybe not for my office, but if I'm in a factory, yes, it makes sense. I want to know where everything is actually happening. If I'm in a big theme park like Disney or another place, I want to know where everybody is who's pinging my access points through either that band and that wristband. Yeah, I think that makes great sense. I think electronic wallets are wonderful, right? So my car, all I need is my phone. I can go shop with my phone, and I can start my car with my phone. I don't need a key or any of that stuff. I think that's fantastic. Other areas, I don't believe are needed, right? And so, when do you actually drive things? So, I've been on the road for 20 some odd years. Back in the day, wasn't known as remote work; it was known as road warrior, right? You had your laptop, you went where you used to go, and I love that now. There's a flexibility, but then when I get on, people like, yeah. I can't go on camera because I'm not dressed. Like, okay, well, it still work. Put a presentable shirt on. Yeah, it's easy nowadays. Not suit and tie. You can use a polo or or a sweater or something simple. And so I think sometimes technology starts invading people in their homes. Smart TVs, I think they're great, but then again, do I want to be tracked for everything I do? I don't. Right. So privacy to me, Keith, is not that I have have nothing to hide. I just don't have a lot I want to share with people. And so having that notion where we need to be tracked no matter what we do because that's how companies make money-they sell that data. I think technology starts confusing that piece. It creates it too complicated. There's kiosks everywhere. There's apps everywhere. I like them, but sometimes I want to talk to a person, and you can't, and that becomes problematic. I think technology also ruined a few things. I love Amazon. I love the ability to order and get stuff real quickly. But now the stores don't have enough. I like to touch things. I like to see how that works. And so technology is great, but then sometimes it gets in the way of how do we manage those pieces. So at work, yep, people expect you to work a lot. Get back your teens message real quickly. All the other pieces. When do you have time to whiteboard? When do you have time to actually process that that discussion? When do you actually have time to actually craft that reply to the email? Is the clock ticking as soon as you received it? Well, that's problematic nowadays, right? And so it is a cultural moment understanding where technology fits in and where it shouldn't fit in, and each of us look at it differently. And so, yes, I live and breathe tech, but if you came to my home, you would say there's not much here.

K

Keith Hawkey 26:57

Yeah,

N

Neil Stein 26:59

And so I like that separation. I like understanding that I don't need to have tech on everything I need to do. Sometimes I just need a simple tool to go do the work.

K

Keith Hawkey 27:08

Yeah, I'm I'm I'm a little old-fashioned in some of those ways as well, particularly around in-person meetings. So many of the meetings that we have today are virtual and digital, and a lot of the times, I think too often professionals can feel like they can kind of skim by a meeting or not really show up or turn off the camera, not be engaged. When you're in person, you've got to be there. You have to be present. Got to show up in your work clothes. You are kind of forced to be a human and be a part of the conversation. So I know time is it's you can get a lot of meetings with a lot of different people around the world virtually, and that's invaluable today. But carving time out to go meet with people. You know, in my profession, I meet with a lot of IT leaders in person. That is, I think nothing is more valuable than that in-person rapport that you gain face-to-face, having to work through and solve problems, especially with a whiteboard in the background. Those are always fun.

N

Neil Stein 28:19

Yeah, definitely, Keith. And sometimes just reading that body language, right? So we talk about the meetings, right? So yeah, tech allows us to be anywhere, but sometimes you miss that out. You and I did meet for lunch, and I think that was a great thing, right? You pick up on a little bit more about that person. Yes, it also pulls people that they actually have to pay attention. So technology does give people an opportunity to be distracted. Yeah, I like to know what my family's up to, but I'm still at work. They're out doing their thing, and if I get a text every 10 seconds and I'm always checking it, I may not be engaged with the people I'm with. So at work, yes, we need to be cognizant of that piece. It's very important.

K

Keith Hawkey 29:04

Neil, this has been an incredible conversation. Where can our listeners find you if they have some ideas they'd like to bounce off of you?

N

Neil Stein 29:11

Well, I'm on LinkedIn. I'm always happy to join, and my contact information is on LinkedIn. I'm currently here at OrthoCarolina in North Carolina, and you know, to me, reach out to me, Keith. I don't know if you post any information on there. I'm always happy to be reached out by text. We talk about tech a lot. I don't answer my phone. I rather get a text, right? And so you know, you know, I'm happy to be reached out that way, or directly through LinkedIn online is probably the the primary way to reach out to me.

K

Keith Hawkey 29:50

With that, do you have any any closing remarks, Neil?

N

Neil Stein 29:52

Yeah, so I've been in technology. I started working on some things that were really new, so the Apache web server back in the late. 90s Tomcat app server. Shortly after, and doing a lot of wonderful things in my career. A wealth of information, and I love to share. But I also love to hear what other people are doing out in the world. And so, if you just want to have a casual conversation, I'm always happy to to have that conversation. I think in in tech, as as Keith and I, we've been chatting about is how do we share that and collaborate on that knowledge, and I think that's very important. You know, to be curious, and I'm always curious what other people are doing. If you're into RC planes and drones, I've been doing that for 15, 20, years as well. And so, if you want to hear about those type of things or get into that? I'm happy to. That's part of me playing with tech and having a hobby. And so, in closing statements, I think more people need to collaborate now, get away from the standard postings, and actually have those conversations. And I'm open to have any type of conversation anybody wants.

K

Keith Hawkey 30:58

Well, well said, Neil. Thank you so much for taking the time to be on the IT Matters podcast, and we will see you next time.

N

Neil Stein 31:06

Take care.

A

Aaron Bock 31:07

Thank you for listening, and we appreciate you tuning into the IT Matters podcast. For support assessing your technology needs, book a call with one of our technology advisors at opkalla.com. That's opkalla.com. If you found this episode helpful, please share the podcast with someone who would get value from it, and leave us a review on Apple Podcasts or on Spotify. Thank you for listening, and have a great day.