Michael Roberts (00:09):

Welcome to the Health Connective Show. I'm your host Michael Roberts, joined by our COO Justin Bantuelle and our company President Scott Zeitzer. For this episode, we're gonna discuss how we've seen AI being used in actual code. AI is gaining traction in the medtech industry, but it needs good data and infrastructure in order to be used safely and effectively. We'll discuss some of the ways we're seeing medtech companies use AI and what foundations are needed before medtech companies can start using AI tools. Guys, welcome. Thank you for letting me corral you into one space again. We've been doing kind of a series of episodes here where we're kind of getting a chance to kind of talk on, sort of what does Health Connective think about various topics, whether it may be, you know, like how we approach prototypes or how we approach different types of projects. So what I wanted to talk about a little bit today is the usage of AI and the practice of actual coding.

Michael Roberts (01:01):

And by coding I mean like actually writing the code that's gonna cause the program to happen, right? Like don't wanna get that overlap with like medical coding, any of those other types of ways that that term could be taken. But so as we're starting to think about like, hey, you're actually writing the code that is gonna drive the software for either a medical device directly, for online systems around medical devices, all of these things need some sort of code behind them. I will speak as the marketer, the guy that doesn't do any of the coding to say that, you know, I even in my corner of the industry, I hear people talk about vibe coding and how you don't need necessarily coders for this type of thing or that type of thing anymore. And I think we touched on this a little bit in the prototyping discussion that we've had previously about like, yeah, you can get something going, but that doesn't necessarily mean you have like something that can drive an enterprise level resource, especially within a very regulated industry and space. So all of that preamble out of the way, let's talk about just overall like what are concerns about using AI in code development. I've touched on 'em a little bit, but I think that there's probably a lot more than what I'm getting at right now. So what are some of the regulatory ethical concerns in the code development process and particularly in a regulated space like medtech?

Justin Bantuelle (02:17):

I guess first off, I've seen some groups be very wary because a lot of these tools require you to provide some pretty deep intellectual property in order to generate what you're looking for. So I've seen wariness to even expose people within a team to these tools so that you can't even put that information in. 'Cause realistically most of these systems are ingesting what you give it to expand out and train on it as well. But I think there are kind of walled garden options coming online more and more that sidestep that. So once you get more into, okay, you're allowed to do it, you've got what you want use, I think a lot of the challenges come down the road. I think initially, hey, you're generating code, you're giving prompts and getting things that seem like they're doing what you want them to do. That's really cool.

Justin Bantuelle (03:11):

The trade-off is if you're not going line by line and learning and understanding what it produced and code review is a very different skill from writing code, and you're reviewing code that you did not write <laugh>, you may miss things. Some of that, especially in this regulated space, could be around security. Did you overlook some kind of security considerations? And that's the nature of humans writing as well. Oftentimes the people writing code are not security experts and that's why you do need these groups to perform security audits. Ensure that the software that was written is up to snuff. But if they have findings, you didn't write the code, you may not be as familiar with where to go to remediate it. Worst

case scenario, you need to do a significant restructure to ameliorate the problem. And when this isn't something you wrote, you're going back, you're re-prompting.

Justin Bantuelle (04:05):

Does it remember the context of what it wrote at all? In my experience, from what I'm seeing, you don't have a lot of that. Maybe in your initial engagement, your initial prompt, you keep cycling on it, but okay, it's a month later, two months later, three months later, that context is gone. <laugh> You can ask it to try to fix this up, but do you have any confidence whether it did it correctly before another security scan happens? <laugh>, right? There's a lot of challenges around what you're producing because the vast majority of software development is not greenfield. It's not new feature development. It's supporting and maintaining and adding to existing code bases. And that's where that contextualization is so essential. And you even have this problem, you could get a very good developer and bring them in on a code base that's 10 years old and has gone through so many iterations and so much of the history, the context of why decisions were made.

Justin Bantuelle (05:04):

If you don't have somebody who has that information, that background, that context, it's extremely painful to pick it up. It's very expensive, it's very time-intensive. You need that knowledge transfer. If the person who initially is housing that knowledge transfer isn't a person but is an AI that produced this, and then, like you no longer have anyone you can consult about any of the decision making process. So any kind of system that's being built like this is intended to be supported for a long time. That's where, as you mentioned at the beginning, hey demos, proof of concepts, just getting something out the door quickly to prove out something, it's fantastic. But once you're talking about an actual robust system that you're looking to support for a long period of time, you're gonna have that challenge. Like that's I think the biggest constraint that I'm seeing people face right now.

Scott Zeitzer (05:54):

You bring up some interesting points, J. Uh, I wasn't even up to, oh my god, somebody else coded it, not an actual person in it. And I was starting in my own head with whose intellectual property is it? You know, that became a big question, I think. Do you feel that most of the companies are getting over that in terms of like, no, no. If our guys are asking the questions, it's our property. Before we even get into, hey, you better make sure you take some good notes. Before we get to that is from an intellectual property perspective, do you think that's a concern?

Justin Bantuelle (06:27):

It's um, I would say it's really case by case. I think some of these organizations that I'm seeing that are more than just like a tech startup, they tend to be a lot more reticent. They have a much wider, I guess surface <laugh> in terms of liability, the risk to their organization as a whole. If they're longer term, they've been more established, they're probably gonna dip their toe in a little more cautiously. I think more of these companies that are really oriented around, we are a tech company, this is the one thing that we do, their advantage in the field is that they're a little more agile. They can move quicker. They have something that they're proving out. They have a runway and a certain amount of funding often and any kind of advantage they can take through acceleration with something like this, I think they're a little more keen to see what they can achieve on that front.

Scott Zeitzer (07:23):

Yeah, it's, it's interesting the, the use of AI for the larger, more established company versus somebody who's like, hey man, I only have this much time. And you mentioned it, I can spin something up a little faster, I can think a little bit quicker. But I do think one of the takeaways I'm getting out of this, you know, sincerely is it's not a panacea, it's a tool. And you better wrap some things around that tool to protect yourself as you're moving forward. Does that make sense?

Justin Bantuelle (07:51):

Yeah, I completely agree.

Michael Roberts (07:52):

You mentioned at the start there the different considerations that need to go into coding and like, you know, the intentions made, the decisions made. You know, I can think of that myself from like, here are the business decisions being made, you know, our end goal is that our revenue model is gonna be based on this. Like this is the most important thing from a revenue model. So the software and all the processes need to be built towards that. But what are some of the other concerns that you're thinking of and some of the other questions that you need to ask when you're starting a new project like this? If you're planning like a major enterprise level tool, what are some of the software questions that have to be asked and answered before you can move forward?

Justin Bantuelle (08:30):

You mean before you like hand anything off to an AI to ask to generate something?

Michael Roberts (08:33):

Or even a coder, you know, like, hey, we've got this decision, we're gonna build this major portal, we're gonna build this major system. What are the the decisions that need to be kind of resolved before you start any of the actual coding itself?

Justin Bantuelle (08:48):

I mean your tech stack, what you're building to, where's it gonna be hosted? What is the architecture, right? Like are you building like APIs? Is this gonna be purely front end as an application? Are you building a mobile app? Like, but those are part of largely the business decision, who they have partnership with, what programming language it's being generated in. Only certain ones can be generated through these tools right now. But those I think are pretty fundamental. Those questions, like any coder who's being asked to start a project, like those are gonna be asked either way. So I don't think that AI is really changing anything about how those questions get answered. I think it's more about the fact that like, okay, a developer's starting now, somebody who has access to this code gen tool, right? And some of them are not really developers by trade or don't have as much background in it.

Justin Bantuelle (09:33):

The emergent issue that I think is coming out of this is if you don't know how to translate, like you said, those business requirements into very specific things that you're looking to generate from a coding standpoint. If you articulate it at a high level like that and you tell it, hey go build this for me, it's really gonna miss the mark. Like you're really not gonna get what you're looking for. You need some degree of coding knowledge to frame up the kinds of things you want it to generate with your code. If you're talking about an actual like fully featured application. For a small like script or something where you're just trying to get it to do one kind of data transformation or like port from one file format to another.

Those things, you're gonna get something pretty good upfront. But you're gonna need to be thinking about every different component of the application and you're gonna be needing to give pretty tailored prompts to those particular components and interfaces between areas of the application. You're gonna need to get deep.

Justin Bantuelle (10:33):

The less experience you have, the more likely you are to really struggle to frame up good prompts and the more you're gonna get, end up getting lost in the weeds. I'm seeing a lot of people even who really know what they're doing, generate this and then say, cool, I generated it in 30 minutes. I got a great scaffolding, but then I had to really drill down for the next several hours to start buttoning this up, getting it up to snuff, like it saved me some time upfront. But the things that you're now thinking about and having to do there, it's kind of the stuff that you would've done even from scratch. So, I don't know, I'm seeing it as being a little bit more of a wash in some ways <laugh> than something that's actually speeding everything up.

Justin Bantuelle (11:15):

I think a lot of developers report that they feel more productive because they're getting something right off the bat. Like they, they have something to work with, but that's not really where the time and effort and the hard part goes. It's in those really niche details, those nuances you had to think through. You generated it, didn't quite do what you want. Why isn't this quite fitting in right? This isn't actually solving the problem I thought it was gonna solve. You face that no matter what. I don't think AI is gonna somehow solve those business problems for you, <laugh>. It's just producing some code.

Scott Zeitzer (11:48):

It, it's funny too 'cause as the owner of a company, so I don't, I haven't coded in over two decades, right. And I rely on you and I'll, I do see, and Michael you'll, you'll, I think you'll agree with me. Like they'll just, so many companies just stick "AI" whatever it is. "I had cereal with AI." And it's like, and I do think one of the things like, you know, as you're starting to think about how you're gonna incorporate AI, which company you're gonna use if you are gonna outsource it is like, alright, what's the big picture plan? 'Cause just 'cause you're stuck AI or you're, you know, keeping it all internal just because you're, the CIO said, you know, we are going to use AI. One of the things that I'm hearing from you Justin, is well how? How are we going to use it? What's a good way to do it? What are the pros and cons of using it? You know, one thing I'm basically hearing is, uh, you might wanna, for Michael and me, we think of editors, uh, you know, people who review the code all of a sudden become much more important because you're looking at it going, what did you generate? You know, which is different. Some coders will be like, this is great, I got started. Other people will be like, <laugh>, I don't even know why I asked for help. I gotta start from scratch.

Justin Bantuelle (13:07):

Yeah. And like you said at the start, it's a tool. It's about how effectively you can utilize it. And I think it is really helping some people. I think what you're hearing promised in terms of marketing is <laugh>, uh, not quite representative of the actual space. So it'll be interesting to see how it evolves and it emerges. But I'm following a lot of actual like dev communities and seeing how people report their wins and the hoops that they have to jump through, the ways they have to learn in order to leverage these tools. How they're seeing the junior developers at their companies struggle in a way that they're not, because like it doesn't just magically solve the problem and replace the developer. It's so new that the studies are very

small scale. They don't tend to have a large population or sample size for these. But it's interesting seeing some of them, like I mentioned just a minute ago, with how productive people report they feel versus measuring how quickly they're actually producing versus people not utilizing the AI. And in those studies it tends to break even. So it's not like AI is hurting anybody. And it is, it does ultimately then come down to personal preference. And there's reason to think that these tools will improve over time, but it's not the remarkable transformation of this industry.

Scott Zeitzer (14:28):

Panacea.

Justin Bantuelle (14:28):

Yeah, exactly. You know, uh, and that's where like you'd mentioned the how, right? How are you using it? That's what I think is so much more interesting than these more generic tools that you can prompt for something. But tools that solve specific use cases within AI. That's where I'm really fascinated within healthcare. Um,

Scott Zeitzer (14:44):

Yeah, it's,

Justin Bantuelle (14:45):

It's uh, seeing the way that certain devices incorporate it to solve a very particular medical condition and help operate this robot or help with diagnostics, uh, those are fascinating to me and I think have real evidence of success.

Scott Zeitzer (14:59):

It's interesting J, 'cause you talk about generating code, but AI can also be very helpful for, as you mentioned earlier, code that's been generated. Code that you have to go back to. I can't tell you how many times we as a company have had to dive back into somebody else's work. It's like, wow, they promised X to get done. We got Y. Or hey man, they did it, but it was five, 10 years ago, we gotta get back in and we don't wanna just throw it out. So what do you think about the effectiveness of, of AI reading existing code and saying, look here, here, and here, or I see holes here. Like doing security checks, et cetera.

Justin Bantuelle (15:37):

I think those are very interesting tools and that's already, that's been a little bit of like code scanning tools have existed for quite some time now. They can identify and report security vulnerabilities. Like, oh, there's a thing called static code analysis, which is like screens existing code base. Uh, there's tools that will try to hammer, uh, applications that are live. And that's where it gets really fuzzy. Like is that AI. Is that not AI? And nowadays it like this generative AI, right? Like gen AI is what a lot of people think of as AI now. That's where it started to get into a more widespread usage. It can give you human-like responses and that makes it feel like it's intelligent, but it's still just an LLM, which isn't really the same thing, but. So the language is very fuzzy on what is and isn't at this point. I, I do think there is value and I see tools that do assist with identifying points of weakness, areas that maybe need to be considered. But ultimately getting computers to do the rote task stuff is something we've been doing forever. That's the whole point of a lot of this, uh,. Like applications and why people code in the first place. At what point that becomes AI versus not AI, I'm not really sure.

Michael Roberts (16:56):

I had a chance to talk with some of the other people in the company. It's not just the three of us, right? Like we've got <laugh>, we've got designers, and programmers and all that sort stuff.

Scott Zeitzer (17:04):

We do indeed. Yes we do. A great team.

Michael Roberts (17:05):

So one of the conversations that we got to have the last time the company was in New Orleans and we had everybody sort of in the same building. I got to ask some of the designers like, hey, as the client, right, I'm, you know, pointing myself as the client is, you know, the marketer requesting something from design or maybe, you know, another client is requesting something from our design team. One of the questions I asked the two of them was, hey, if I came to you with something, I had used an image generator to spit out and say like, hey, I've got this idea, spit out this thing and then I bring it to the design team and I go, this is what I'm trying to get at. Like, I used 84 prompts before I finally found, found the right one. And I think I'm kind of close, but like what do you think? Like how can I use this to like get to where I'm going? I'm curious, like we've talked a little bit about this from a prototype standpoint, but like would you view that you, Justin, the program lead <laugh> for our team, would that be helpful for you to see? Does that help get you closer to like, hey, we understand what your requirements are? Does that help the client think through what their requirements are better? Like I'm curious how how you would respond to that sort of approach.

Justin Bantuelle (18:10):

I think there absolutely can be a benefit there. So first off, the first draft of requirements is never actually the requirements for the final product. There's so many things that you overlook, you hand wave like, and it's just the nature of us. Like we tend to oversimplify problems until we have to actually dig down and do that. It's just how humans are. But a lot of times something doesn't work the way you thought it was going to, you forgot something. And I think a client being able to confront some of that and observe the friction as they're trying to produce that, uh, it helps them think through and get closer to a final state of what they need. I am not troubled by the idea of somebody having done a deeper dive into what they need before they come to me.

Justin Bantuelle (19:03):

And I think having something that's visual. So often, if we're talking what you imagine in your head, the way you interpret what I'm saying versus what I mean, the way that I think I effectively communicated something and I forgot to contribute this one important component and it completely changes the context of how you took what I was saying. In this like verbal and then even in this written format, right? Uh, we're operating off of two things that we're hoping are in alignment. Once you produce something visual, there's something a little more concrete at that point that forces both of our understandings into a closer alignment. I think that's why we do mockups, prototyping. That's the purpose of those things, right, is to ensure that we're on the same page and that we've thought through all of the considerations. I think there's nothing but benefit to that. I've often had people come to me with very messy mockups and they're like, I don't want it to look like this, but does this help you understand what I'm trying to say? And it absolutely does. So this doesn't, like if you generate something like that and it understands a couple of clicks and it does some things, it has some interfaces, you're starting to see the different

components and how they're supposed to interplay. Even if it doesn't really, even if it's very, very janky it's still communicating something meaningful.

Scott Zeitzer (20:23):

Yeah. It's, it's interesting 'cause for someone who's a programmer leaning into this podcast, I think they're getting a good feel for it. For someone who's not just kind of from a big picture AI integration, I do think the idea of like, hey, I came up with a logo or a look and feel graphic, you know, uh, interactive team, do you think?

Justin Bantuelle (20:41):

Correct. Don't make it look like this, but this is the idea I'm trying to communicate. It's now. Right. And not an idea in your head. It's an idea on paper in some way.

Scott Zeitzer (20:50):

And I'm pretty sure that the designer would come back with like, hey man, it's a start. I can ask questions like, did you like the color scheme? Did you not like the color scheme? I'm taking a look at the overall, like my eyes are going up to the right. Is that what you were looking for? And I, I'm oversimplifying, I'm the king of oversimplification. And I do think like for code, you know, you're able to get a feel for, am I talking about the same thing here? I see that you're coding for X, Y, and Z. Uh, your business reasons seem to be this. Like, it does help you get, wrap your head around the larger project. I do think it's, it's not, again, I'll say it over and over, not the panacea yet. It's not like science fiction where you go and you're like, make me this.

Michael Roberts (21:34):

Earl grey, hot.

Scott Zeitzer (21:35):

Earl grey tea, hot.

Michael Roberts (21:37):

<laugh>. Just go.

Scott Zeitzer (21:38):

Alright, thank you all Star Trek fans. Yeah, <laugh>,

Michael Roberts (21:40):

That's like the one reference I remember <laugh>. So

Scott Zeitzer (21:44):

<laugh>

Michael Roberts (21:45):

To expound upon that idea a little bit, what are the benefits and dangers then of somebody going like, I was able to say a few things and I got a version of it. Shouldn't it always be this easy? Isn't that what

you're gonna do? You know, just with a little bit more, like I, I see like there's positives that can come from it and especially like me, like I do like benefit from like visual results and, and I think a lot of people do, right? But do people get locked into like, well it's just this easy? Is is that a danger that we run into?

Justin Bantuelle (22:14):

That's absolutely happening in the field right now. You have all these CEOs saying that they're gonna fire 50% of their developer workforce because of AI. Then you have other CEOs saying that is absolutely moronic and those companies are gonna fold or be rehiring at a much higher rate than <laugh> before.

Justin Bantuelle (22:31):

So some of it is just, let's see how it shakes out. Like I'm just one guy giving my opinion on this based on what I'm seeing, right? But there's a lot of companies that are leaning very, very hard into this and committing. There is absolutely a sentiment among developers of frustration because they do have their managers saying, well I did this, it was this easy. Like what you're saying is literally happening to people on a day-to-day basis. And I think that is a somewhat myopic view to take. I think that without understanding that the hard part of development is not the lines of code produced, but how effectively it solves that problem. How much you're marrying the business needs to what you're developing, how you're translating that in an effective way that meets user needs that you're considering these more intangible user experience pathways. There's a lot in there that is different from "build a site that lists out this information."

Michael Roberts (23:28):

Mm-hmm <affirmative>.

Justin Bantuelle (23:29):

So I think the deeper you go, I think the more complex you get. I think again, going back to maintenance, what you build today is not gonna be what you need in a month. Like maybe it's close, but you're gonna need to make changes, right? You're gonna need to be able to update it, support it, remediate security issues. Uh, there's a whole slew of things that go into keeping a application alive long term. And the simplest part of it is that upfront getting started piece of it. So I think there is a trend of downplaying the importance of what developers actually bring to the table beyond just writing the line of code. A good one really does need to interface and understand how they fit within the organization as a whole and the needs of the business users, the stakeholders, all of that. And it's not being held by these companies that are selling the AI and promising the ability to do that. They're not left holding the bag if their promises are unfulfilled.

Scott Zeitzer (24:30):

<laugh>, you know, it's, it's funny bringing that up as you were talking about all this, you know, about the adoption of AI, firing, hiring, et cetera. I think about outsourcing, you know back 20 years ago when the idea of everything's gonna be moved offshore. Because it's so much cheaper and they can get it done wherever they are.

Justin Bantuelle (24:53):

Right. Developers being interchangeable cogs in a machine. Yeah.

Scott Zeitzer (24:56):

No problem. No problem. And I, and I remember a lot of medtech companies just saying like, yep, we now have medtech company, India. And I remember like after about six months to a year of trying it, you were like, yes, the hourly rate is a lot less, but now I've got the following things that I have to pay for. Whether it was management, whether it was redoing it, et cetera. It wasn't, again, the panacea. Is there a lot of offshore application development work? Absolutely. And I think we'll find our new balance. Do you agree with that? Like over time we'll find our new balance of how much gets moved over, outsourced, et cetera to AI. And how much again? No, you're gonna need some onsite developers, fellas.

Justin Bantuelle (25:40):

You're, you're not the first one that I've seen make that comparison. I've seen it.

Scott Zeitzer (25:44):

I thought it was brilliant. I thought I had it all myself.

Justin Bantuelle (25:46):

Well, no, no. I mean I don't see a lot of people saying it, but I think it really is a good point. I think it's, uh, the fact that you're observing that and then people deep in development spheres also seeing that is a very salient point. They're saying that it, they're anticipating the same sort of backlash, a lot of people here. Where just assuming it's gonna magically solve all your problems is absurd. The outsources that I think work more effectively tend to be the ones that actually end up costing a bit more upfront, but create fewer of these problems down the road. And that's where I think there really is an analog there. With the outsourcing, like I mentioned context again, right? Like does the person who built your code know? Like when you are working with these giant companies with these big contracts where it really is just they're swapping people out. Like the person working on your project this week might not be the person next week. And that's where a lot of it happens. It's not that any specific developer offshore, in the US, wherever is an idiot or anything. It's that like the actual organizational structure is not empowering these projects to succeed and these developers to succeed.

Scott Zeitzer (26:51):

Absolutely.

Justin Bantuelle (26:51):

And the more you try to cut corners, cut costs, the more you're gonna start seeing that friction. You're gonna start suffering for it. And it's the same here, where if you think that this is just gonna solve all your problems, you're absolutely gonna stumble into a lot of the same issues where long term, it's really not gonna work out for you. You didn't plan for maintenance and support of what you needed.

Scott Zeitzer (27:11):

You know, one of my biggest takeaways that I got from the outsource movement, and it, everything swings back and forth, et cetera. And I think the similar thing will be about AI. It's like, look, there are certain things that are great to outsource. There will be certain things that are great for AI. I know that the successful companies that outsource actually have a lot more budget on managers to make sure they get what they need to go have those conversations with the company they're outsourcing to. Uh, I should say offshoring too, because there's gonna be extra costs involved, the time delay, et cetera, et cetera. And I go back to AI like, hey man, not the panacea. There will be certain things that it's great at,

there, it will get better, but you're gonna have to spend more dollars doing X, Y, and Z. And if you ignore the cons, embrace the cons I guess is where I'm getting to, embrace them. Okay, yeah. Here's where we've got some issues. What are we gonna do to overcome that if we're gonna use AI, right? Who are we gonna have to bring in to do that?

Justin Bantuelle (28:13):

And then is it worth it to do it at that point, right? Because then you're having very honest assessment, right? It's like it could be a win for you. You could anticipate that you're a little bit ahead at the end of it, but it's not like an obvious all pros, no cons. There's no option that has that.

Scott Zeitzer (28:26):

<laugh> No such thing, man. No such thing. Right? So my takeaways for everybody would be, it's not the panacea, but understand the pros and cons and then work hard to understand how to accentuate those pros and minimize those cons.

Justin Bantuelle (28:39):

Absolutely.

Michael Roberts (28:40):

One of the discussions that I've been hearing in, at a particular marketing group that I'm a part of, and it's really interesting because it, they really are willing to kind of like extrapolate on an idea. Like, hey, if, let's just pretend like AI takes over major, major portions of your business. What are the differentiators and how does any company stand out in any given space, right, if you're all drawing from the same source? And I think so much of what you've talked about in terms of like how the programmer, how the team lead, how whomever it is like is actually keeping like, hey, we wanna match our coding to our business objectives, to are we really solving the problems as effectively as possible? So there's, regardless of which tool or who is the one implementing the work, there is still like very key needs around overall vision for something and then overall expertise to ensure that that vision like actually plays out.

Michael Roberts (29:33):

So I, I think that's one interesting component. Another interesting component that I saw just in a discussion around, it may have been like when Salesforce was reducing some of their customer service team, is hey, all these systems are getting more expensive now that like some of these different systems are in and they've got that foothold. Now the price to do these things are going higher. And it's funny because just the little bit of playing that I've done on, hey, can I get this little simple application to get going? I ran outta tokens like stupid fast <laugh>. And then it was like, do you want to pay for more? I'm like, no, I don't want to pay for more. I'm out.

Scott Zeitzer (30:09):

No, no. I thought it was gonna be magically done. It's done. Right?

Michael Roberts (30:11):

Good. I thought, you know, pixie dust and all that kind of stuff. So there are just interesting complications that are coming out of it. And I think one thing that, to kind of like play on the other side

there is like, we're not anti-progress in technology, right? Like artificial intelligence can be a major, major thing here, but there are a lot of things that the tech company CEOs don't want to say because that's going to lower the value of their product. It's gonna like, it's not gonna be as easy to adopt it and everything. And, and they are trying to overcome this like big mind gap of like, how do we actually use this thing? So just lots of interesting, yeah, things that are happening in this space right now.

Justin Bantuelle (30:48):

One thing to kind of build off of with what you just said there, AI, you tell it, build this thing to like address this need, it's gonna do it. It's not gonna stop and say, that seems a little weird. Are you sure that's actually what you need? Like, did you wanna do this instead? Most of the people in the industry are doing this to solve that problem. Right? It's, it's that critical thought component. It's the part that actually is the intelligence that like we bring as humans that the AI is not. The AI is just following directions, <laugh>. But yeah, it's, it's not gonna bring any of that critical thinking at that like business requirement layer and that's where things get messy.

Scott Zeitzer (31:25):

Yeah. I, I don't think we, I mean I hope we as a company do not, we're not pooh-poohing AI. We embrace it. It's just utilizing it as best it can be utilized. And looking at it continuously, because AI is going to change. It's gonna get better too. And we're gonna have to all of us, not just us at Health Connective, but all of us are gonna have to look at it with open eyes.

Michael Roberts (31:49):

Absolutely. Thank you for letting the mind expand during this half hour or so. It really is fascinating to kind of, to address these conversations at a bigger level of just like, can I do it or can I not? But like what are the implications? What are the real.

Justin Bantuelle (32:04):

The tradeoffs or the consequences.

Michael Roberts (32:06):

Yeah. Like I think that these are important conversations that need to be had like in every single area where new technology's rolling out, like. And it's rolling out so fast. It's so funny. Like I'm, you know, talking to my son about this stuff and it's like, you know, he, and he's getting ready to go to college soon and it's like learning to think through these questions are gonna be the things that are gonna be impacting our lives so much going forward. So guys, thank you. I appreciate it.

Scott Zeitzer (32:29):

Sounds good.

Justin Bantuelle (32:30):

Thank you.

Michael Roberts (32:31):

In our episode, we discussed how medtech companies are currently using AI in the code development process, what regulatory and ethical concerns exist, and what medtech companies need to consider.

Thank you to our viewers and listeners for joining us for this episode. For more on the Health Connective Show, please visit hc.show for previous episodes and Health Connective as a company.