Transcript Details
This is a transcript of an educational program. Details about the program and additional media formats for the program are accessible by visiting: https://reachmd.com/programs/the-convergence/Michal-Nedoszytko-PostVisit-ai/54781/

ReachMD
www.reachmd.com
info@reachmd.com
(866) 423-7849

"You don't have to know programming to create solutions," says Dr. Michal Nedoszytko, Creator of PostVisit.ai

Dr. McDonough:
Welcome to *The Convergence* on ReachMD, where innovators and physicians explore the technology transforming medicine. I'm Dr. Brian McDonough, and today, we'll be discussing how a practicing cardiologist placed third out of 13,000 people in Anthropic's hackathon. Joining me to talk about what he did, and his new tool, is Dr. Michal Nedoszytko. He's the founder of PostVisit.ai and an interventional and general cardiologist at Hospitals of Europe in Brussels.

Dr. Nedoszytko, it's great to have you here today.

Dr. Nedoszytko:
Thank you for having me today. It's really an honor.

Dr. McDonough:
And it's an honor to have you. It's really exciting.

You placed third in Anthropic's hackathon. As I mentioned before, 13,000 people competed. You're a practicing cardiologist. You were coding in hospital hallways, between patients, and on planes, using Claude servers. First of all, how do you pull this off? And when did you even get to the point of saying, "I'm going to try to do this?"

Dr. Nedoszytko:
The decision to participate in the hackathon was actually completely random. I've never participated in such a competition. The story behind it is that I already had a trip to San Francisco planned. I was supposed to meet someone who'd launched a healthcare startup. A few days before departure, I just saw this ad, and I said, "Why not?" I'd already coded for many years. I created software for healthcare.

And the idea behind PostVisit.ai was something that was actually from three years ago; I'd created a completely different system based on AI called PreVisit.ai. It was an AI agentic intake system for patients that gave the doctor a summary. But the post-visit was something that was quite difficult to build in Europe, and I stalled with it. And then, when I saw this hackathon ad, I said, "Why don't I showcase it?" But I had no idea how many participants there would be, and I absolutely had no idea the success and everything that would ensue.

Dr. McDonough:
As we move through our conversation, we're going to talk about what you built and how you looked at the post-visit. A lot of physicians think about what's happening during the visit, and you were focusing on after the visit as well. And as a family doctor, I know how important that is.

But talking about the actual act of what you're doing, first of all, for those who understand, we know coding can be very complex. But to do it at the level you did—and it basically was one week, and you were barely sleeping—paint the picture of what you were doing.

Dr. Nedoszytko:
The sleep thing—actually, interventional cardiology and jet lag helped a lot. But I've been coding all my life. Throughout my medical career, and even in med school, I've been creating software for whatever has frustrated me in the medical pathway.

But recently, the tools that we have at our disposal have just made things so much easier, at least to prototype. It's hard to describe. I will, of course, talk you through the whole process. But personally, I always had this itch to transform whatever I could optimize in my work inside working software. But the hackathon and the model that Anthropic released—the Opus 4.6—it was just a beast. What this model could achieve in a very short time span was nothing short of remarkable. So, I think that the stars aligned.

© 2026 ReachMD

Dr. McDonough:
We're going to go through the whole story, but when you were doing this, you probably were just going through the exercise and enjoying the challenge. When did you get to the point, “This might actually win?”

Dr. Nedoszytko:
I had no idea about this until the moment that they actually announced it. There was this form to apply for the hackathon, and the only thing that it said was just, “Pitch us your idea, and you'll be among the 500 participants.” We had no idea that there were 13,000 applications. It was five days before my departure to San Francisco, and I got an email in the middle of the night: “You've qualified for the hackathon, and you can start coding in two days.” And I was just like, “Oh, okay. I cannot code now.” The rules clearly stated, "You have exactly seven days to create a working demo and a demo movie," and I just started creating it about 24 hours later—the moment that they had announced the rules about what you can do, the license requirements, repo, and that it should be public.

And what ensued after this was also an incredibly interesting experience because they created the community of 500 builders that were interacting with each other during this whole hackathon. This was a worldwide online hackathon. So, it happened in a virtual community, and people were just pitching their ideas and helping each other. Most of them were actually non-technical; I think maybe 30 or 40 percent of people actually did have an engineering background. They have created, using Claude Code, the solutions that they’d had in the back of their heads for a very long time.

But my idea— I had absolutely no clue. At the end, they announced six people who went into the straight finals, and it was 20 minutes before announcing the places. And then I placed third. And I was still in my hotel somewhere in San Francisco like, “What's happening?” And everything that followed after this was nothing short of remarkable.

Dr. McDonough:
We were flattered, actually, because after you won, I know your LinkedIn dramatically exploded briefly. How do you even think of a way to cut back on all the LinkedIn messages when you're getting so many of them?

Dr. Nedoszytko:
Currently, you don't have to know programming to create solutions—or at least prototypes of solutions—for something that you can use personally. Even throughout the process of creating PostVisit.ai, I have created separate tools to help me a little bit.

So, LinkedIn is one example. The moment that the tweet was posted by someone, it went completely viral. I needed to actually turn off all the notifications because my phone just kept vibrating all the time. And I had 2000 invitations on LinkedIn, and I needed to understand who was writing. So I created the scraper—and that took me 20 minutes using Claude Code—that connected inside the browser, looked throughout my request history, and actually categorized it and showed me how I could actually deal with this response.

And another cool example was when I was actually creating the video. I really thought that the video was only going to be for six judges, but somebody posted it, and then it just went completely viral on the internet. I had some lines that I wanted to say—for example, “I’ve done thousands of procedures in this room.” That wasn't actually me; I'm personally quite a humble person. But Claude Code actually suggested that it would augment your credibility in the face of the judges that you have some authority and so on. I had those lines that I wanted to speak, but I couldn't remember them. It was just one afternoon in the cath lab that I could do it. So, I created a teleprompter, and it took me 10 minutes. Claude Code basically created an application that worked in Chrome. It was a teleprompter that I could read out of my laptop. My wife was holding the camera. So you can just create tools or even instant applications that can work really well.

Dr. McDonough:
It's amazing. I want to go back to your origin—medical school. Now, when I was in medical school, I thought I was pretty bright. And I love broadcasting, so I would spend evenings at radio and TV stations pouring coffee for people, doing whatever I could to try to get my foot in the door. And I thought that was pretty dramatic. But you weren’t just studying; you were building tools way back. You would see a problem, and you would come up with an answer, whether you were doing virtual exams or you were letting students exchange ideas. How does your mind work like that? Was it always there naturally?

Dr. Nedoszytko:
I think it was always there. I started coding around the age of 15 or 16, creating simple websites and whatnot. When I was actually prepping for med school, I created these preparatory tests for med school, and they also went online. Many people were using it. So, I always had this parallel thing: whenever something needed optimization, I created software for it.

Med school, for example, was the pre-Facebook era. I was the head of the Student Scientific Society of Pediatrics. It was a really fun time. So, I had created this social website where everybody could post their profile, interact with each other, create scientific projects, and so on.

© 2026 ReachMD

There were many things throughout my medical career. When I was a resident, I hated doing paperwork for lab results. So I had created this script that autogenerated PDFs so that you could just put a stamp on it with what was necessary, and then everybody started using it. So it grew into the full EHR with discharge letters. Then, we had this problem with duties that were organized just by post-it notes, so I created a module for management of medical duties.

Then, when I went into interventional cardiology, I created a full EHR for protocol notes so that it would be easier. You select the artery and the lesion, the hardware is basically filled in by technicians, and then you just select the stent size pressure. And it creates full protocol notes, which is really necessary for reporting statistics.

Dr. McDonough:
All of us who have lived through the nightmare of the electronic medical record and trying to figure out how to work in order sets, we sit around and complain, and maybe we come up with a way to help get rid of some hard stops. But what you are doing is recreating things and using your imagination—the idea of gamification for medical education, making it competitive, and then it works. What about the average physician out there who maybe knows a little bit about technology, wants to do more, and realizes that things are starting to change and they're a little more available than they were? Where do they even begin? If somebody says, “I could never do that,” what's your advice for them?

Dr. Nedoszytko:
I think that every physician has some big idea in the back of their head and wants to create it. Most physicians publish papers. I was always publishing less; I was creating more code. And before, you had to go through the documentation and learn libraries and some programming language. But since about three years ago, things have started to shift.

When AI became mainstream, you could use tools. First, it was Visual Studio Code with some plugins, where you could just discuss it and it created code. You could review it. But progressively, it started shifting. You don't even have to actually review the code; you just interact with your program or application, and you actually have to understand the code less.

Andrej Karpathy, who was the head of autopilot at Tesla a while back—he's an incredible brain and a big influencer on what's happening currently with software—he said that the best programming language that you could learn in 2025 is English. And that's absolutely true, with a small caveat. I actually consider Polish more efficient because there was this study by the University of Maryland and Microsoft where they categorized languages in terms of efficiency of interaction with large language models. Polish actually came out first. Then there was French, and English was actually sixth. Long story short, right now, you can just interact with your tools and the most common tools that you can use.

And one thing is called the model, which you could just compare to the level of intelligence that the system expresses. But another thing is the tool—the way that you actually interact with the model. The hackathon was actually organized for the first anniversary of Claude Code. Claude Code is a tool made by Anthropic. It was designed by Boris Cherny. Claude Code started out as a terminal tool. If you see geeks and hackers doing something on their computer, they're just typing something on this black screen, and the characters appear. Terminal is a very simple way to interact because it's very fast; you can just type, and things will appear. But before, you needed to know the command-line interface, and you needed to know what commands work. Here, you just type code, and then you just talk with the terminal and ask them what you want to do. They have also launched a desktop version. And if you need a server, it can actually configure a server for you. So everything that was required before—going to go to websites and getting a server and whatnot—it can actually guide you through it and it can do it much more easily. So, for me, the terminal was a natural environment where I was always working because it was very efficient.

But fret not—you actually do not have to type, because right now, the most efficient way of interacting with LLMs is voice. And it's not actually talking to it and looking at the waveform, but actually dictating. So I'm using, for example, a tool called Wispr Flow. Wispr Flow was very natural because you could talk in many languages at the same time. I used Polish, French, Dutch, and German. I speak multiple languages. But it also works very well on your iPhone; you can create a virtual keyboard where you have this button, and you dictate. If somebody types quite fast, they might do about 140 words per minute. But if you talk, you generate 300 or 400 words per minute.

And what is important for LLMs and for your project to succeed is really thoroughly explaining what you need. The models are really intelligent, and they can do very well, but it all depends on how this idea will be conveyed to the model and how well it will understand it.

Dr. McDonough:
Somebody once told me that if you're going to do anything like this, the train's moving; just get on it. I get the impression you're saying, “Get the tool, and then just start.”

© 2026 ReachMD

Dr. Nedoszytko:
Yeah, the best way of learning is actually by doing. I made the decision to participate in the hackathon. It's not really about talent; it's just the decision to try. Currently, AI allows you to prototype much faster. So if you had an idea, and at that time, you needed to get the team ready and it took months, if the idea failed, you lost a lot of time and resources. Currently, you can just prototype it locally.

One thing to mention and what is important is also that we still need programmers and software engineers because there is a big difference between demoing something and actually creating a production-ready software, especially in healthcare. In healthcare, there's a lot of guardrails; there's privacy, there's data security, there's compliance, and there's a lot of legal. So, for example, the hackathon demo had all the steps necessary to have HIPAA compliance and so on, but it is not a production-ready program. PreVisit.ai, which I built three years ago, absolutely, yes—although building in Europe is a completely different story.

But let's come back to how somebody who has an idea and no technical background can actually start doing this. So, you can either just go with Claude Code and then tell them, “I need this database for patients with chronic heart failure, and I have this subset, and I have access to this kind of data. I can get it out of the EHR, but also, I have some people working on the next file.” And you can just put it on your computer and give paths to this file. It will start analyzing it, and you will say, “I would like to review the data. I would like to see it. And can you make a browser outlet that will allow me to interact with it?” And you just start going and iterating. This is a way to go ad hoc and start doing things immediately.

But if you want to build something that is viable, my process usually is to be a product manager. So first, you discuss very thoroughly—this is how I did it with PostVisit.ai—you create this thing called PRD, so a product requirement document, and then you ask Claude Code or any other tool to create this document where you want to describe what you actually want. “How do you validate that this works?” “What should be the output?” “Who are the users?” Just make sure that the model gets it. And the moment that you are at this stage that you think, “This is good.” You just ask, "Alright, so now, prepare the plan—how you would implement it.”

Before, models were struggling with something called a context window. They could just do these kinds of chunks of work until they lost their memory, they went out of context, and they went out of sync. So, you needed to divide this work into certain steps, which is still the case, but the context windows are becoming much bigger. So, you can actually delegate a lot more work in one chunk.

But first, you create the PRD, and then you tell it, “Give me your proposition on how you would actually implement it. Give me a plan.” And then it says, “Okay, I will start with this step where I will give you mockups and create the database.” And then you will validate if this is fine. And then you go step by step saying, “This is good.”

To give you an idea, PostVisit.ai is a quite complicated system. This is really something that requires connecting to electronic health records and wearable data. It needs to interact with the patient. It needs to actually build on top of very technical things that the model provides—sequential thinking, tool use, HIPAA compliance, and so on. I worked for two or three hours actually discussing with Claude Code how would I like this system to be created. And then, when I created a quite sufficient and long PRD, I said, “Go to code link.” I thought that it was going to take a day or two. That was my experience before. And 40 minutes later, I came back to the computer, and it said, “Hey, I'm ready." "What?" And I checked—It was actually quite good. Of course, there were errors; I still spent a few days after that to polish it. But the speed was nothing short of remarkable.

And there was something profound, I think, that happened in engineering somewhere around December 2025. The models were struggling a little bit. They went off the rails after doing something thoroughly complicated. But I remember I had this break between Christmas and New Year's Eve, and I had this project that I'd been doing for a very long time. I tried the new model, and in five minutes, it created this functionality test and so on. And I was amazed how fast it went. I thought, “Maybe people are just on vacation and they have their time off, so the servers are faster.” But then, people started resonating on the internet that there was something that happened in the models. And it was not only Claude; it was also OpenAI models that became just so much efficient. So there is some kind of revolution that's been happening since about three months ago.

Dr. McDonough:
You're listening to *The Convergence* on ReachMD. I'm Dr. Brian McDonough, and I'm speaking with Dr. Michal Nedoszytko about his work.

And I'm moving on now to what I think is really exciting. People build AI, and a lot of the AI that's been built helps doctors during visits. But you focused on after the visit. Why is that the gap you chose to address?

Dr. Nedoszytko:
Well, it came a little bit from the clinical needs. Eight years ago, I moved to Belgium. And in Belgium, the system is much more saturated when it comes to pathology, I would say, than in Poland. There's much bigger density of hospitals. And before, I spent much

© 2026 ReachMD

more time in the cath lab, and here I started spending much more time in the outpatient clinics.

The first need that I had—it was actually three years ago, when the first large language models were released—was that the visit itself might be a little bit inefficient. When you interact with the patient, there's those repetitive questions which come all the time. “What medications are you taking?” And patients sometimes don't remember. They say, "Oh, my wife knows. This red pill." And this becomes a nuisance because within the limited time that you have interacting with your patient, you could make better use of it. And instead of actually trying to wait until he calls his wife or trying to correct the name of the medication until you finally agree upon it, you could actually go into the diagnostic part and make better use of your neural network.
I try to experiment with large language models, and I have created PreVisit.ai. The system talks with the patient before the visit in a very polite and empathetic way and asks all the pertinent questions. I started out with cardiology, first visit, comorbidities, medication, risk factors, and so on. Then it provides you with a medical-grade summary. But the summary shouldn't replace the anamnesis. It just gives you a script so that you could interact with the patient. You can use some parts of it in the EHR, but the goal is to increase the quality of your interaction with the patient.

And what I have observed actually is it was not only the real-time gain and difference, but the quality has improved tremendously because the patients talk at home, where they know what they take, they have support of their family, and moreover, they are better prepared for what will happen during the visit. We are already getting straight to the point, and the visit has just become so much more efficient. That was the idea. And I actually created PreVisit.ai three years ago. We have integrated it with many systems in Poland, Belgium, and Greece, and it works really well. We already have 100,000 pre-visits that are working.

The only caveat is actually that you need to tap directly into the workflow. You cannot provide an accessory or a secondary tool. There is a big gain in efficiency, but not many people grasp that immediately. We needed to tap immediately into online agentics and electronic health records so that it's directly integrated.

But coming back to PostVisit.ai, another problem is that when patients leave the doctor's office, they usually forget a lot of information, like, “What is Mediterranean diet?” “Why did he actually give me this medication?” “What are the possible statin adverse effects?” “What does it mean by muscle pain? Where?” “When should I come back?” “What lab work?” “When should I do the CT scan?” “Should I be fasting?” Sometimes, a lot of patients try to call you, and you’re stuck with all this demand.

I wanted to create this tool a long time ago. But the problem is that in Europe, unfortunately, creating any sort of solution that implies the diagnostic part of AI is incredibly burdened with red tape. So I concentrated more on a pre-visit administrative tool. I felt that when I came to the US, there has been a complete other principle where you have a big boost, and there's not much red tape. And yes, of course there's HIPAA and PHI, and GDPR also works here. But people start actually creating. The amount of startups that are blossoming currently around healthcare is astounding.

I created PostVisit.ai out of a clinical need. I wanted to close the loop, not only so that the patient can actually understand every part of the visit and their medical record, but also so that they could actually get access to the doctor in a more efficient way. If you have adverse effects after propranolol and you want to know if it's actually propranolol, or maybe you should reduce the dose or whatnot, you can actually ask this question, and it goes back to the doctor. And there's another LLM that will help the doctor actually analyze whether the patient needs to go back, whether the patient can just have his dose reduced, or if you can just reassure him. This makes this part of the post-visit much more efficient.

Dr. McDonough:
As a family doctor, I can see where your motivation was, because especially with, for instance, heart failure, when people go home, things can really fall apart. They get confused about their medication. Maybe they're gaining weight. And the next time you see them, they're back in the emergency room or admitted to the hospital. So, I think what you're trying to do is get past that—walk me through it. You have PostVisit.ai. You've developed this tool. Does the patient have a microphone they're talking into? Are they on their computer? How do they get the information essentially through the tool?

Dr. Nedoszytko:
Yeah, so basically, I needed to find a way to get the most accessible data inside the context. And on the one hand, you can try to access the EHR, which is incredibly fragmented, and in most cases, actually, it's virtually impossible to access it. But if the patient actually has their own tool, then they can collect the data.

So one way is reverse-AI scribe. Long story short, AI scribes are systems that listen to the discussion between the doctor and the patient, and then they create protocol notes. So these are mostly doctor-facing applications. And I thought, why not create a tool like this for the patient where the output of the transcript is actually more centered around patient needs? And of course, it needs consent.

© 2026 ReachMD

So I thought, all the information that you get during the discussion with the doctor, you will have it, you won't lose a word, and you will have one thing that is relevant to the context. By context, I mean the data that you put inside the model. Then you can access the data that the patient has at their disposal, like wearable data or Apple Health data, if possible. You can upload lab work or any other documentation that you have in paper or PDF format. And because medicine is a science based on evidence, the most important part of it is actually putting in the clinical reference because we won't let the model actually do the clinical judgment if it won't be based on actual medical data. So I wanted also to give this part where you can actually upload guidelines or connect to existing services that can provide real clinical guidelines and reference. And based on this, you could get the output that the patient needs.

Dr. McDonough:
You're really empowering patients to be better partners with you. This is really giving them a chance to team up with you. That kind of changes your role as well, too. It expands it, doesn't it?

Dr. Nedoszytko:
Previously, you had patients that came with a Dr. Google diagnosis. I personally adore the patients who come in with Dr. ChatGPT. I have a lot to talk about with them. And they are so well prepared, and for me, giving the patient a tool that will let them understand their health is actually going to improve you as a doctor as well because you will become a better specialist not going through the noise of the data that is irrelevant and trying to steer the patient towards the better way. You are actually a clinician at that time. You're treating the disease, which is identified more easily.

So I personally believe that healthcare should be optimized in that way, and that we should try to reduce the noise that surrounds the diagnostic and treatment part by enabling both the patient and the doctor. It should go both ways.

Dr. McDonough:
And big picture, as you're already seeing it—and I know you've thought this through—do you think it'll make your job ultimately easier or harder? I think I know the answer, but I want to ask you.

Dr. Nedoszytko:
I think it will absolutely make it easier. It depends on what you like to do, but for me, I absolutely love treating and identifying. We doctors are special kinds of neural networks. We train all our life to identify the patterns around our patients and categorize them into clinical categories and diseases, and then we just use the tools that we have at our disposal. And if we eliminate the administrative burden—the noise—this will allow us to better do what we are trained for.

Dr. McDonough:
How will it change your practice? If all your patients were using this in your cardiology practice, do you think your success rate and your metrics would be better? Do you think they'd be helpful here? What are your thoughts?

Dr. Nedoszytko:
Let's say there are two elements. The first is the part that I think is the most relevant nowadays, and this is the administrative workload. Because if you see currently the elements that we complain the most about as doctors in advanced countries, it's basically the administrative workload, the burden that it creates, and the rate of burnout that actually ensues after this. This will help tremendously when it comes to focusing more on the clinical part and, I would say, bring back joy to medicine.

The second part is basically the efficiency and how healthcare can actually improve in terms of outcomes. So imagine that we don't know why hospitalizations take so long. It's not because we don't have the capacity to treat the patient, but sometimes, going through all the data and identifying it just takes so long, and the doctors don't have the bandwidth. So you could dramatically reduce the hospitalization duration and also reduce the hospitalization rates. Let's take the example of chronic heart failure, right? We all know that in order to reduce early hospitalization, you need to follow the patient's weight. You need to detect the signals that will tell that the patient will soon crash and will require hospitalization, which will mostly take a long time.

So this kind of patient empowerment will also allow this loop to close. We have a lot of problems in post coronary treatment. There's a low adherence to antiplatelet medication. And the amount of thrombosis that we get after discharge is not really that high in Belgium, but it is a significant problem. And it can help with compliance if you have a tool that is friendly, that is easy to access, and that presents the patient with the data in the most digestible way. It can tremendously help with maintaining compliance and the readmission rate and basically adherence of the patient to do a whole follow-up.

Dr. McDonough:
With innovation, there's a great deal of responsibility as well. And ticking off some things from a list, you build something fast. And you're able to do that. But how do you factor in patient safety? Or if a patient misinterprets something from PostVisit.ai, and all of a

© 2026 ReachMD

sudden, when they misinterpret it, they make a harmful decision—how do you guard against those things?

Dr. Nedoszytko:
So one very important thing is that current large language models are mostly probabilistic, and as high as this probability could be, we need to make sure that in medicine we still adhere to the same principles that we were adhering to for ages, which is deterministic, which are predictable outcomes. So that's why an essential element is the doctor in the loop that always validates and takes responsibility.

The pace of innovation in engineering and software is absolutely blazing. It's incredible how fast those tools evolve. And medicine, unfortunately, will struggle a little bit to catch up with this, which I don't necessarily think is a bad thing. We still need to validate clinical trials. We still need to adhere to evidence-based principles. And in engineering and software, you can validate if a solution works by doing something called unit testing. And if it fails, it fails, and then we just create the other code, which is not the case in medicine because here we deal with human life and quality of life. There are guardrails and principles, and what we need to do is basically try to convert our clinical practice, principles, and workflows inside something that will be predictable and inside something that will actually be useful.

In Europe, it's quite difficult because medicine and all the healthcare applications regarding AI are currently classified as the highest. And if you actually want to create a similar solution, then you need to go through the whole process of screening, audits, and explainability, which is an incredibly difficult thing in AI models. But you need to also adhere to the principles of patient safety, of traceability, of how you actually share the data, of who can share what kind of data—the patient needs to give consent to certain elements. And the doctor needs to give consent to certain elements. So healthcare is much more complicated than this. Demoing something or creating a small tool is incredibly rewarding, and it can actually push discovery further, but in order to create something that works on a population scale and actually helps the doctor with the clinical decisions is a completely different story. And it needs to be created by people that know what they're doing, who have clinical experience, and who are responsible for creating clinical solutions already.

I think AI will accelerate that. Clinical trials can be actually created with better scope and better efficiency. We have already tried using PreVisit.ai in selecting patients for clinical trials, and actually, the groups of patients that you get are much better than if you would do it manually. Of course, you always do manual screening after this, but you can just create that in rare diseases. I think that's a very modern solution.

And at the hackathon itself, among 500 people who were the builders, at least three solutions concentrated around rare diseases. Some of those were actually about generative data, and some of those were about looking for patients with rare diseases. For example, there was even an interventional cardiologist from Japan who created this taxonomy system for better explainability of medical coding in Japan.

Dr. McDonough:
I'm interested in a couple things you're doing and how it really plays out. As a physician, you were the lone coder working on the plane at 30,000 feet, and you were alone doing things between patients. But also, you've recognized the importance of medical teamwork because whatever you develop has to be studied for best medicine, it has to be team-based, and you have to collaborate. And when you talk about collaboration, coding, and building, are we failing our medical students or our residents? Should we be teaching them about building and coding and encouraging them? Because one thing's clear to me: the insight you have as a practicing physician is helping guide your decisions in what you're going to do with AI. Are we failing them by not getting some of these medical minds more involved at an early stage?

Dr. Nedoszytko:
I wouldn't go so far as to call it failing, but AI is the current outlet to creativity. It is absolutely necessary to give future doctors the tools and an understanding of how they could use them to better shape their work. Currently, as Boris Cherny, the creator of Claude Code, said, programming is mostly solved. What is not solved is scaling the software onto the production levels. But if you have an idea, you can just create a working solution for that and do it as fast as possible.

If I recall the problems that I had during my medical studies, I would create so much more. It sometimes took a long time. The electronic health record for interventional cardiology took me about three years to actually code during the nights and during the shifts. Now, I think it would take a few weeks, and maybe somebody could help me with scaling into production levels and securing it. But I think that it's absolutely compulsory to add programming—well, I wouldn't even call it programming—AI coding, AI generation, and AI creativity inside the curriculum.

There is this term called Vibe Coding that was actually pitched by Andrej Karpathy, which, at the beginning, was quite trendy. But I don't

© 2026 ReachMD

really like this term because vibe coding means that you just absolutely don't look at the technical part of you have created—how it works under the hood and so on. And while it's really cool and groovy to just vibe and talk to your AI and create some nice program that will work in healthcare, you need to understand the process. And that's why I said that it's quite important to first lay down a document with the requirements, both on the user interface and on the backend—so the things that interact with your database and servers—and on the security part, because this is something that you will have to validate every step of the way. So small things, yes. But if you build bigger projects that actually deal with a lot of data in clinical settings, then this is also a part that needs to be done.

I've talked a lot about AI for about three or four years. I am present at many congresses and scientific meetings, and I talk about how AI is reshaping the world. And I think that it's only in the last year that people started actually to grasp the potential of it. And currently, the potential they realize is more in a general way. So, you have your ChatGPT on your phone, and you can talk with it, and the outputs that it produces are just "wow" because you can discuss your taxes and you can discuss your current problems. But the real deal actually is to go into this creative part and start transferring your domain expertise—which was actually the theme of the hackathon—into some kind of a working solution.

Dr. McDonough:
We have a lot of physicians who listen to this, and I know they're probably thinking, "I want more of this doctor. I want to hear more. This isn't enough time." Where can physicians follow you and reach you? Is there a way they can learn more about what you're doing and learn more from what you are doing?

Dr. Nedoszytko:
Yeah, certainly. My LinkedIn went into normal mode, so I can respond freely. I'm also on Twitter. If you want to reach out, don't hesitate. I will be very happy. I also think that I will be in the US more often because I quickly realized how big the opportunity is, and I was just astounded by the speed and the creativity of the community here. I came here three weeks ago. I was supposed to stay two weeks. I needed to extend my stay, and I didn't know anyone. And right now, I just interacted with the sharpest and brightest of Stanford. It's incredible how fast you can just meet people and validate your idea, and this is the center of the universe when it comes to AI development.

Dr. McDonough:
We'll be lucky to have you. You're talking about being with the best and the brightest; you're very bright yourself, and yet you're humble. And I think that is one of the attractions. You're open to talking to people, and you're showing physicians they don't have to wait for IT departments or vendors. They can do things themselves. I think that's a very powerful message.
And I want to thank you for spending the last hour with us. I really appreciate the time, and I know how you are in demand. And I thank you for sharing your journey and where it's going, and obviously, we're all going to be following you and hopefully connecting here and there in the future as well.

Dr. Nedoszytko:
Thank you so very much for having me.

Dr. McDonough:
You've been listening to The Convergence on ReachMD, where innovators and physicians explore the technology transforming medicine. To hear about other technological advances shaping the future of healthcare, visit The Convergence on ReachMD.com, where you can Be Part of the Knowledge. Thanks for listening. We'll see you next time.

© 2026 ReachMD