Shane McGraw: Hello, and welcome to today's SEI webcast, "AI Engineering: Ask Us Anything About Building AI Better." My name is Shane McGraw, Outreach Team Lead here at the Software Engineering Institute, and I'd like to thank you for attending. We want to make our Q&A as interactive as possible today, so you can submit questions in the YouTube Chat area now, or if you're on LinkedIn or Twitter, use #askusAI and we will get to as many as we can.

Our featured speakers today are Dr. Rachel Dzombak and Dr. Matt Gaston. Rachel leads our Digital Transformation Team for the SEI AI Division. In her role, she works with organizations to realize the capability of artificial intelligence for mission outcomes. Prior to joining CMU, Dr. Dzombak was an Innovation Fellow and Professional Faculty Member at the University of California Berkeley.

Matt is the Director of the SEI AI Division and the Founding Director of the SEI Emerging Technology Center. He's also an adjunct professor at the CMU Institute for Software Research.

Welcome, Matt and Rachel.

Rachel Dzombak: Thanks, Shane.

Matt Gaston: Good morning, Shane. Glad to be here today.

Shane McGraw: Yeah, it's great to have you guys. We have lots of great questions already coming in over the last couple weeks in preparing for the webcast, so we're going to get right to

them. So I think the first one we kind of got to ask you guys is, to kind of level set, is, "What's the difference between AI and AI engineering?" so I'm going to point that one to Matt first.

Matt Gaston: Well, thanks, Shane. That's a great question to start with. So yeah, we're here today to talk about AI engineering, and it's important to set that in the context of the field of artificial intelligence. So AI has been around for a long time. Usually considered to be the field of study where we're looking to figure out how to make machines intelligent, for some definition of intelligence, and AI is a very broad field. It's been around for a long time. I think 1956 at Dartmouth the term was coined, and AI includes a lot of technologies and has been through a lot of waves over the decades.

Today at CMU, at Carnegie Mellon University, we often refer to what we call the CMU AI stack to look at what AI includes, and it includes everything from the low-level computational resources necessary to realize AI capabilities, up through sensing and processing of data and information, to machine learning, and I'll come back to that machine learning concept in just a second. And then on up through reasoning and planning, human-machine interaction, autonomy, and includes a wrapper of policy and ethical concerns, and AI is really about the science of all of those things and how to incorporate ideas across that whole stack into building intelligent systems.

So let's talk just a little bit more about machine learning. So machine learning is one piece of AI, and really, I like to think of machine learning as a way to create AI capabilities. Traditional software systems, you know, software developers are programming specific rules, in a sense micromanaging the functionality of a system. In machine learning systems, algorithms that learn from observation or learn from data are used to create a model, and that model can do some set of reasoning or inference on its own.

The important distinction between machine learning and traditional software engineering is that there isn't that micromanagement process, right. There's an algorithm that learns from the

training data and then the model itself just sort of behaves based on what it learns from that, those observations, and then inside of machine learning and really a big focus of modern AI, modern machine learning, is this concept called deep learning, where we've taken an old concept, a concept that was created in the '80s, maybe even the '70s, called neural networks, which require a lot of data, a lot of computation, and so in the last two decades we've realized that the internet has provided an opportunity to produce lots and lots of data available for these types of algorithms, and we now have the compute available in a very democratized way with the cloud that allows us to train very, very large neural networks to do some amazing things, like recognizing the damage to buildings after a natural disaster, from satellite imagery.

And so AI is really all of that stuff, right. Remember that full CMU AI stack, and AI engineering is really an emerging field. It's not--it's been around for a while in different incarnations, but it's really an emerging field that recognizes that there's this huge power and promise of these AI technologies. There's great applications, there's great things that can be done with these technologies. But we need to improve the discipline around how we do that in a reliable and responsible way.

The best way I've heard this described, AI engineering as a concept, is from Michael McQuade, the former Vice President of Research at CMU and member of the Defense Innovation Board. Michael describes AI engineering as "doing AI as well as it can be done," and so it's not so focused on individual capabilities or particular algorithms but around the qualities that we want these AI components and these AI systems to exhibit. How to make them trustworthy, reliable, responsible, et cetera.

So I think that's a--maybe a gentle introduction to AI and the concept of AI engineering, and Rachel, you might have things to add.

Rachel Dzombak: Yeah, thanks, Matt, and thanks again, Shane, we're--I'm excited to be here today. Really excited to answer questions from the audience, so please do keep those coming.

The only thing I wanted to add on was just that there are so many people looking at individual components of this across the AI research community and their focus on specializing in those specific aspects. So focusing on model development or on data management processes, and I think a key distinction for AI engineering is we're taking this whole systems approach, trying to understand how do these elements intersect and how do they come together to drive towards the outcomes we're looking to see? That's the challenge right now with AI is the implementation, particularly in contexts where ambiguity is higher.

You know, it's one thing if you are trying to be recommended a restaurant and the stakes are low if you get that recommendation wrong, but in higher-stakes scenarios, so, you know, in Matt's example of seeing a building after a disaster and understanding the damage to that building, there's a lot of factors there, a lot of variables that come into play, and so figuring out what does it mean to achieve that, to get towards that, that outcome, and use these systems in the way that they're intended, it requires us not just to look at the model that is the basis of the system but all of these individual component parts and where they come together.

Shane McGraw: So we just had a question came in from Twitter and I think's relevant to this section, so I'm going to leave it open to either--to Matt or Rachel. This was from Mark Nassing. "What's needed to build an AI engineering mindset in a team?

Rachel Dzombak: So I'm happy to kick that one off, because Matt knows it's one of my favorite subjects. I think that one of the biggest elements is some humility, and a curiosity for learning, because the field is evolving so quickly that I think there's a tendency for people to think, "Oh, well, Person X, they have all the answers. They have all of the expertise that's needed," and it's an impossible goal right now. It's an impossible goal for someone to know absolutely everything because the field is too big and the techniques are changing all the time. The leading practices, there's new ones coming out every single day from different parts of industry, and so you need a team that can flex and adapt to where you are in your process. You also need a diverse team that can see the problem spaces differently to pay attention to things like potential unintended

consequences or different impacts that a potential system could have. With AI especially, in removing human decision-making, it opens you up to all sorts of potential consequences that can be really detrimental to human lives, as well as human prosperity.

So I think that that mindset of, "We are learning how to learn together," that we are thinking about this and are constantly going to need to keep updating, evolving our mental models, we're never done, that part is critical. But also understanding that one person doesn't have all of those answers and you need to work amidst your diverse team, which is much easier said than done, and that is a piece that I think is critical moving forward.

Shane McGraw: Rachel, let's stay with you for--just for one more here while we're kind of setting the stage here. "How does AI engineering relate to other engineering disciplines?" and that was from Heidi.

Rachel Dzombak: Yeah, absolutely. So, you know, I'm an engineer by training and work with a lot of computer scientists, and I've found it interesting in this field that there's a lot of competition between the two and people kind of drawing lines between them. In my mind as an engineer, engineering was--engineering disciplines evolved as an applied science. You know, how do we leverage technology, how do we leverage math and science to solve problems that matter? That could look like developing products, processes, systems. That's what engineers do. We try to approach the "How?" side. How can we reach this place we want to get to? How can we reach this desired outcome that we have or our customer has? And I think that's the mindset we're taking with AI engineering.

It would be--we can't look at the system development without also understanding the context in which it's going to be implemented in. We're learning from a lot of other engineering disciplines around what does it mean to build rigor? What does it mean to understand systems? There's great examples to be taken from civil engineering, from mechanical engineering, about how do you start to problem solve in a way that's context-specific? So we are learning from all of those

disciplines and trying to build towards that, and, you know, when we define this field of AI engineering, we say it's integration of systems engineering, human-centered design, computer science. We're pulling those elements together because it's what's needed. You can't just view AI from a single discipline and expect implementation to work perfectly. You need this multidisciplinary approach, and we view our roles not as having answers either but as being curators and helping to weave threads between those different disciplines and bring them together.

Shane McGraw: Great. Before we go to Matt with the next one, I love keeping eye on the chat, just, you know, seeing that we got people from Uganda, Kenya, Mexico join us, so we got a truly worldwide audience it's great to see. So keep letting us know where you're from and questions you have.

So Matt, let's go to this question for you from Joseph asking, "How do engineers navigate tradeoffs among AI system qualities, such as between robustness and scalability? What would a person in that decision-making role use to determine tradeoffs?"

Matt Gaston: Yeah, this is a great question, and I think still a largely open, open question in--as far as the engineering discipline is concerned. But what I'm excited about related to this question is this idea that we call beyond accuracy. Let me start by talking about traditional ML ops and traditional and machine learning operations and building of machine learning models. The focus is usually on what we call model evaluation, right. So I've got a very precise, very specific task, like let's use the simple example of classifying cats or dogs in images, and I can measure accuracy of that task, right. So I have data that has cats and data that has dogs, and I run it through my model that I've trained and I can measure the accuracy of that model. This is the large focus of the work that gets done in building of machine learning capabilities, a worry about this model evaluation concept and how well it does on the specific task.

To get to the actual question and this idea of beyond accuracy, we're keenly interested in methods that, rather than looking at model evaluation and the precise, you know, classification task, it's really important, and Rachel just said this, it's really important to think about the business or mission or operational context where the system, where this model, will be deployed inside of a larger system, and that context can actually drive metrics, and those metrics are that balance between robustness and scalability and accuracy.

In a really great example that I'll credit Martial Hebert, the Dean of Computer Science at CMU with, there's a very common machine learning task called image segmentation, and so what image segmentation is is given an image, how do I identify which pixels belong to which object in that image? So there might be a road and a car and a tree and a mountain in the background. Pixel by pixel in that image, can I classify what object that pixel is part of? And if you go into the academic literature and look at image segmentation papers, the metric they use is overall accuracy, and so how accurate am I on a pixel-by-pixel basis?

Now, if you think about that, in most real-world applications, this probably isn't required. It's probably not required to have every single pixel perfectly accurate, and so this is going beyond that accuracy measure and thinking about, "Can I trade off computational resources? Can I trade off robustness? Can I trade off accuracy to build the mission capability or the operational capability that I'm most interested in?"

Rachel Dzombak: I can add one thing onto that, which is I think this is where our work in AI engineering has a lot to learn from other engineering disciplines as well, because it's about how do you make smart design decisions and how do you track the impact of those decisions over time? Right now, a colleague of ours, Jay Palat, he likes to use Mary Shaw's definition of discipline growth, and says that, "Right now a lot of AI's in the crash phase. We are trying things out. We're doing a lot of customization to get to that mission outcome, that business outcome, but we're not at the place of standardized, repeatable processes."

You know, a developer can run their program multiple times and get different answers, and it's a question of, "Well, why does that happen?" and I think to get to that place where we are understanding what the tradeoffs are and how to navigate them, it's going to take us as a discipline, you know, some continued reflection around those intentions of, "What are we trying to do, why are we trying to do it, what's the context in which we are making this tradeoff, when does it make sense to prioritize one design consideration over another?" and starting to share across those so there's lessons to be learned from the discipline.

You know, I think that's a focus of a lot of our work at the SEI is learning from others and trying to bring together those lessons learned and help people unpack the choices they made, whether implicitly or explicitly, and I think especially in this craft phase, a lot of times the choices we're making are more unconscious. We're just moving through or hacking it together to get it to work, and it's a matter of refining that process and thinking about, "What was my motivation there? Why did that technique work? Why did I make this decision?" and getting clearer on that so then it can be shared and repeated by others in the future.

Shane McGraw: Great, Rachel. Let's stay with you for this one. William sent in, "How are your foundational efforts and AI engineering leveraging leading-edge research in the overarching field of sociotechnical systems?"

Rachel Dzombak: Yeah, I love that question. Thanks, Shane. But I think, you know, historically so many engineering disciplines in computer science have been deeply technical and have wanted to wish away the context parts, so social parts of systems. But, you know, I think Matt and I are both big believers that when you're trying to solve a problem, it's typically a problem that a person has. You know, people have problems. They are trying to make things happen in the world for a reason, and so our technical systems need to be built with those end users in mind.

So we are leveraging a lot of previous work in sociotechnical systems, a lot on behavioral insights, so why would someone--how do we reduce the inertia for someone to make a decision? How do we reduce friction for people to navigate and gain understanding? What does explainability look like in this context? What parts need to be explained?

You know, we've talked a lot about how I might not know how every single part of my car works, but I trust that it'll drive me from here to the grocery store, and so what is enabling that trust and how do we build systems in a way that are allowing people to have a similar level of trust? That cannot--if we just built the technology and didn't think about the people side of it, we wouldn't be successful, especially when working in high-stakes applications like the work we do in national security. So instead, we have to be drawing on the psychological elements, the behavioral elements, all of those pieces, and going back to the early question on education, that means we also need to be training our engineers to have what I call social-technical fluency to be equally at home on the social side and a context side as they are on the deep technology side.

Shane McGraw: Okay. Let's go from--one from our live chat from Nermaine asking, "I suppo--" and I'll leave this to either one of you so you can jump in. She asks, "I suppose the AI architect is part of the AI engineering group, so what's the difference between an AI architect and a business analyst?"

Matt Gaston: I'm happy to take a quick attempt at this question. I think that the AI architect serves as a translation between that business analyst, right. The business analyst presumably has deep knowledge of the domain where a particular application might be operating, a lot--bring a lot of that context and knowledge about the domain and the business operations, and I think the business analyst and the AI architect work hand in hand to understand how AI is going to affect the business or can affect the business and what are the right things to do and the right questions to be asking about the AI technologies that they might be trying to incorporate? And so AI architect maybe brings a little bit more understanding of what's possible with AI. Business analyst brings a little bit more understanding of the business and what is needed, and that combination, right, that teaming of those two roles, I think, gets us to the right questions that need to be asked in the designing and deployment of these types of technologies.

Rachel Dzombak: The only thing I would push back on is that it could be two roles. It could be a person who has purview into both spaces, you know, and this notion that no one has all of the answers. Finding people who can live in both worlds, who understand a bit on the architecture side, also understand the business side, that, finding that ground, finding people who can cut across, super important for teams right now, especially because oftentimes people in those two roles speak different languages. They have different terminology, different nuance terms, and you need folks in the team who kind of can--not necessarily that are expert at either, but are comfortable flexing into roles as projects need them, and I think increasingly we're going to see more of that. I heard someone coin the phrase multi-potentially, of people who have these kind of multiple pathways or diverse backgrounds to be able to live into both worlds.

So I really think about it as what are the jobs to be done to understand the system that you're trying to build, what all needs go into it, versus kind of saying, "We need a person to fit specifically into that one role"?

Shane McGraw: Great. Thank you for that. We're going to go to Matt on this one. John had sent in, "Since most deployments fail, however, what can the work to define the AI engineering discipline offer now or in the near-term, for AI systems now in deployment that could prevent them from failing in deployment?" And let me know if you need me to reread that, Matt.

Matt Gaston: No. I've got it, Shane. This is a critically important question. I presume--I think you said the question asker is named John.

Shane McGraw: Yeah.

Matt Gaston: I presume John is familiar with this Gartner study that's been publicized and shared through various venues recently. About 85 percent of machine learning projects in industry never make it into production. I don't know that I would say fail, but never make it into production. So the important first part of trying to answer this question is to talk about why we think they might fail, and so I'll point to some work that's based on the AI Incidents Database, which is a database of AI failures from--I think it's run by the Partnership for AI or at least sponsored by the Partnership for AI, and the Center for Security and Emerging Technologies at Georgetown just recently did a study of that database, and the conclusion of that study was a categorization of the failures and why these systems fail, and there were three core failures. Core failure classes.

The first is specification. So the desired intent, the desired mission application or business application, wasn't specified correctly or specified in the appropriate context, and so maybe the AI system that was built didn't actually solve the problem that the teams were trying to solve. The second is robustness, and that gets at testing. Maybe the system worked in the lab environment, in the development environment where it was being built, but as it gets integrated into a larger system and then put out into the wild to operate, the testing processes weren't robust enough to capture all of the use cases and edge cases that the system might be faced with, and then the last one is assurance, and the assurance there is once a system is deployed, how do we know that it's continuing to operate the way we want it to be operating?

And it turns out that environments shift and data sets shift and, you know, to use a statistical term, distribution shift over time, and that affects how these systems behave. So once you deploy a system, the context in which it's operating can actually change. Like think weather conditions, right. If I've got a machine learning system that can see cars on the road, and weather changes, right. Rainy conditions, snowy conditions. The road goes from usually a darker color to a lighter color when it's snowing. That's an environmental shift that can have an effect on these types of systems.

So specification, robustness and assurance, and then I think those are the three things that there are recent developments on how to better capture the application domain, the actual intent

that we want these systems to operate, and that gets to what I've already talked about in beyond accuracy. Thinking beyond just how accurate can I build a model but how am I going to deploy this model, what are the types of questions that are most important that it does well on, where can it not do as well, and where is that tradeoff space? So that's for specification.

There's lots of work going on in testing, and there's an increasing understanding of how these systems can fail on their own without intervention, but also how these systems might be manipulated, which is an interesting topic all in itself that would maybe require an entire-- another webinar like this.

And then assurance. There are--there's a growing focus, both in the--in sort of the machine learning science community, as well as the machine learning practitioner community, about monitoring these systems and how to calibrate these systems appropriately so that the systems themselves can actually know when they don't know something or know when they've stopped being able to perform as well as they can, and so there's any number of a variety of techniques and emerging tools that can be applied to those types of problems.

Rachel Dzombak: One thing I would--

Shane McGraw: Agreed.

Rachel Dzombak: --would add there is just that, you know, Matt and I are big fans of learning from failures. Not just casting them off of, "Ah, this didn't work," but being intentional to look at, "Well, why? Why did they fail? What was at the root of that?" and of course the different areas that Matt just laid out give lenses into it, but I think oftentimes it's super nuanced. You know, I think there's a lot in the news right now about people pulling investments out from AI applications in healthcare, and I think the articles leave a generic answer at times saying, "Well,

they just didn't work. Doctors didn't adopt them." But there's probably a lot more to it, and I think it's a place where I would love to see the community keep doing work to understand that--understand what actually happened in those situations or unpack it a little bit. You know, we have a--just a quick plug--we are doing a symposium for AAAI, which is one of the leading AI conferences, and we would love to see people submit some case studies on failures, trying to understand what went wrong and why did those things go wrong?

And the flip side of that is finding oppor--case studies where successes happened and understanding what enabled that success in the spirit of amplifying positive deviance. You know, was it just that the model was so fantastic? Was is that you achieved that notion of a very specific use case, you had assurance and you had robustness, or what else actually allowed that to happen? Was it the team? Was it the problem? Was it the context? Was it some combination? I think it's a space where, you know, our focus on implementation, we need that. We need more people looking at what actually goes on and what are those enabling circumstances, enabling factors? Because to the earlier points on sociotechnical systems, the answer is going to lie in both of those categories, and I think that's one of the main drivers that will move this forward is being able to articulate those and make them more--bring them more front of mind.

Shane McGraw: Okay. Another question from our chat, and I think it's related, and really the reason why we're here today, and we'll stay with you for this one, Rachel. Peter wants to know, "AI has brought a dynamic black box to software. How can we guarantee quality, security and safety?"

Rachel Dzombak: Sure. So, you know, I think, one, I would challenge the notion that you need opaque or black box models--we're trying to move away from that language--but that you need to--your system has to include an opaque model. There's a great article from MIT that looked at when there are more transparent models being used, how they have become more explainable, more adoptable, and more understandable in that context.

But I think guaranteeing safety is a false paradigm perhaps to aspire to, and this is just my opinion. We want to have trust, but, you know, my example of driving to the grocery store, there's no guarantee that I will make it to the grocery store in my care. My brother always says, when I say, "Fly safe," he reminds me the most dangerous part is getting in the left to get to the airport, and because there's just--there's risk involved, and I think that's the part where we have to better understand "What's our tolerance of risk, when will we use it?" and I think challenge the assumptions about how systems are being built today. Because there's a reason huge uptake hasn't happened in all use cases. It means there's more to work on with the technology and how we're developing it, how we're designing it. It is certainly true that there's more--that a lot of what happens in AI is opaque. We don't understand why exactly it works. The engineer in me views that as a challenge of, "Wow do we get better, how do we grow our understanding? When is that true, when is that not true?" to kind of explore it a little bit more.

Matt, I don't know if you have anything to add there.

Matt Gaston: Well, Rachel, I think your focus on risk and understanding the risk framework in which--and the risk context in which these systems are being deployed and used is critically important, and like the engineering qualities, right, tradeoffs between scalability and robustness, there are tradeoffs that you can take on in the risk context as well, and so I think risk thinking, risk analysis, is a big piece of understanding how these systems can be deployed.

It's also important to note that--so I wanted to mention that there is work in formal methods, right. So presume some of the folks out there, because we're the Software Engineering Institute, are very familiar with software engineering ideas. I don't want to get into explaining really what formal methods is, but formal methods is a method for making guarantees about systems. There is ongoing research and recent breakthroughs on applying formal methods thinking, formal verification, to deep learning systems, to neural networks, to make some guarantees about the boundaries of where those systems might go out of their operational domain, and so that's pretty promising research still on the science side of things.

From an engineering perspective, and again, channeling our software engineering roots, in traditional software engineering you have a lot of tools for introspecting, analyzing, unpacking the behavior of a software system. These are tools like static analyzers that actually go through the code and analyze it for failure modes, for bugs, for violations of secure coding principles, those sorts of things. We do a lot of work in those areas at the Software Engineering Institute. We also dynamic analyzers, which are the, you know, sort of run-time analysis of systems, and then even reverse engineering tools, and I think there's a great need and this is an area we've identified as critical to AI engineering, and some work from the explainable AI and interpretable AI communities on how do we build those same sorts of tools for these models? And like Rachel said, I think it's important maybe to get away from this idea of black box or even that language, but just there's a model there and how do I understand what it's doing and how do I analyze it, and how do I test it on its boundary conditions and how do I find those failures or bug modes?

And so there's, again, interesting developments in this field. There's some tools out there already that you can find in open source or in the public domain for starting to get at the analysis and understanding of these types of models.

Rachel Dzombak: The other thing I would add is just that I think in this day and age we have to continue planning for uncertainty. You know, certainly COVID-19 threw everybody a curveball and helping us, you know, and forced us to look at the uncertainty, the unmodeled phenomena that exists in the world, and that we can't anticipate everything that's going to happen and build them into our models. I think it's hard, you know, there's a tendency for us to want to plan for it and control all possible scenarios, and stepping back to say, "Things are definitely going to emerge that we don't know what to deal with. There are definitely factors that will emerge that will compromise our systems, that will make them less robust. How do we start to plan for that? How do we build that resilience into the design so that we can tolerate these system shocks in a different way that versus hoping and crossing our fingers that we've thought of everything that could potentially happen?"

Shane McGraw: Okay, great. I'm going try to combine two questions from our live chat. Now I'll start with you, Matt, but feel free to kick it over to Rachel. First one was, "Do you have recommendations around a strategy to formulate and AI center of excellence for a large enterprise?" That's the first question, and then the second, somewhat related, "Is there a template that exists for doing AI maturity assessments?"

Matt Gaston: Ah, yeah. So I'll make just a few comments here. We're being a little bit careful about maturity models and maturity assessments at this point because the space is large and complex, but we do have some sense of technology readiness and things that you can be doing to assure those thing--to assure these systems.

And then I think the first part of the question, which is more about organizations, Rachel was just--just published an article or a blog post on AI readiness for organizations, so I think Rachel's best to talk about that.

Rachel Dzombak: Absolutely. Yeah. I think there's a lot to be learned from other centers of excellence. You know, I think over the past decade we saw innovation centers within enterprises emerge quite strongly, and a lot of those weren't successful or people weren't satisfied with their investment, because they didn't have a clear intent of what they were trying to do with it or how it was going to connect to their organization. They didn't have the reporting structures in place to allow idea flow to emerge. Oftentimes those innovation centers became separate from organizations. There was kind of the real business and then the people who were exploring things over here, and I think there's--that's something to pay attention to. You know, when you're thinking about AI centers of excellence, you know, is it that you have a separate center in the organization or do you have one person from each team that is kind of--both has a foothold in a specific function of the organization but also is exploring where and how AI can connect?

Having those different dialogues and ensuring that there's communication flow across the organization and starting small. You know, I think that's a part of right now there's--the failure rate is high and it's kind of your classic challenge with prototyping, where you've put a huge investment in and then it doesn't pan out in the way that you think it was, it was--it would, and it affects the bottom line and it affects people's perception of this technology that you're trying to bring into the system, and so starting to think about, "What areas in my organization are right for AI? Where do problems exist that we're trying to solve? Is there an experiment that we can start to do in the next six months with a small team?" versus, you know, and also I--the first question of being, "Where do we have data in the organization that we currently can use in new ways?" that's a critical first question and I think one that people say, "Oh, oh, we'll find the data." But it's the start--the main starting point that you have to have and then it's a nontrivial lift to pull together that data to, you know, really kickstart everything else that you're trying to do.

So some level of experimentation, some level of thinking about, "How does this fit into the broader organizational strategy? What behaviors am I wanting to incentivize?" Is it get AI rolled out at all costs? Probably not. It's really about how are you leveraging AI smartly to drive outcomes? So how does it affect--connect to the organization's strategic goals?

If for ever--whoever in the chat asked that question, we also have public office hours that Shane can link to at some point, or we'll put into the chat, and I'm happy to bounce those around more and different ideas you're thinking about if you want to utilize those office hours. They're open to anyone. We're happy to talk about partnership opportunities or work that's going on in your small business or large business or whatever that--or your research ideas, your case studies. It's been an experiment that we've been running for I think six months now and has been really interesting to see who we connect with, everyone from educators who are starting to think about, "What does it mean to develop AI engineering curriculum?" to small businesses that are struggling with resource constraints, to large organizations that don't know where to start or have some great leads and want research to support what they're doing.

Shane McGraw: Okay. Great answer. Thank you for that, Rachel. Let's go back to Matt for this one. Kim has sent in, "What AI engineering efforts do you find particularly promising?"

Matt Gaston: Well, we've talked about--we've talked about some of them. You know, I think the expansion of testing beyond just evaluating model accuracy is really important, and we're seeing more and more of that. Even from a security perspective, right, we--I mentioned earlier that machine learning, modern machine learning systems in particular, have an interesting set of security concerns around how they might be manipulated in certain ways, and this can be, you know, they can learn the wrong thing at training, they can do the wrong thing at inference, or they can possibly reveal information that you don't want them to reveal, and there are tools now for understanding those types of security policies in a particular context and testing for those, and so I think the focus on testing not just that model evaluation but at system integration and then into operations and monitoring is one of those, those big areas, and then I--I'll sort of make the meta observation.

I've noticed, and not just because of, you know, the Software Engineering Institute and our perspective on the importance of AI engineering, but out there in the wild, right. As I go to different venues and conferences and have conversations with folks, the shift from, the maybe you would call it hype around AI, right, the craze around AI and what AI can do to transform business and all kinds of applications across a broad variety of spaces and domains, the shift in the narrative is now, "Well, how do we do this right? How do we do this in a responsible and reliable way?" and I think that as a development, right. Starting to shift that narrative and think about, you know, "There is--there is this amazing power here, but what are the right questions we need to ask? How do we build this in a way that we can know it's operating the way we want it to, and how do we monitor it and have it give information to its user community about how it's making decisions and how it's continuing to operate?" That narrative shift from a meta perspective is quite promising, from my personal perspective.

Rachel Dzombak: I want to agree with Matt and amplify just the nature of conversations that are evolving right now. I think we're seeing a lot of conversations around the power dynamics that AI systems create, the values that are imbedded explicitly or implicitly into systems, who is involved in making these systems, and who does it affect, and what is the disconnect between those two things? I certainly have a lot of excitement that those conversations are growing rapidly and I hope that they are, you know, it's not just our echo chamber that those

conversations are happening in that they're getting across, but I have a strong level of belief that that's the case and we're going to see that more and more in the coming years.

The other thing I find really promising are the tools that have gained a lot of traction that have helped with these questions of transparency and explainability. So the classic one is data sets--or data sheets for data sets and model cards for model reporting. Those are small behavioral interventions of having folks document, "Where are you getting your data from? What are the attributes of this data source? Why did you choose it for these applications?" and instantiating those processes, making that, establishing them as part of the development pipeline I think is super critical for transparency and repeatability, all those aspects that we are seeing, and I think more and more those types of small, different, you know, they're high-leverage point opportunities. It's a small change that can lead to a lot of impact. I think that more and more is what we're going to see as companies figure out, "How do you navigate these different situations that we're in and how do we build towards more rigorous process on the creation of these systems?"

Matt Gaston: There's a really cool example that I want to mention here, because I think it ties all of this part of the conversation together. Just yesterday I learned about this idea. In traditional software engineering--and this has been done all over industry and government--there are these things called bug bounties, right. Can you find a bug in my software capability, my software system? And companies will pay, will pay you lots of money to find a bug in their technology. It's a way of sort of open sourcing the testing of their systems.

Just yesterday I learned about a challenge that was put on by Twitter they called a Bias Bounty Challenge, and what they--so Twitter has a capability, and I don't know a lot of the details. I'm not an expert on how these systems work within Twitter, but they have an algorithm, a machine learning algorithm, that identifies the key part of an image that's put into a tweet, and they auto-crop the image based on that, and they put that technology out on the web and said, "Hey, researchers, people that are interested in this problem, anyone out there that can help us, find bias in how this algorithm is working and how this machine learning algorithm is doing what it's doing," and they got a lot of great results and many results that they didn't expect to see, many

types of bias that they didn't--they couldn't think of as themselves, and you can search for that out there. I don't have a link ready for you, Shane, to put into the chat, but just a really great example of bringing engineering concepts like bug bounties in traditional software systems to this machine learning world and bringing, you know, taking a sort of large-tent approach to understanding how to explore, analyze, test, these types of systems?

Rachel Dzombak: And also, that's a great example of planning for unintended consequences, as it would've been easy for kind of Twitter to feel confident, say, "We don't have any bias." They'd be lying to themselves, and instead they said, "We're sure there's bias in here. Even after all the work that we've done internally to make this systems, why don't we put it out there and allow ourselves that vulnerability of sharing and putting that out there?" because ultimately it'll make their work stronger on the other side of allowing that exploration to happen from their user community.

And I think the other, the piece that Matt talked about that made me think of is we're hearing also more and more of organizations that are being proactive to have the whole kind of unintended consequences workshops to engage their kind of user community to think through potential scenarios, potential use cases, and all of that work is quite powerful and quite important, even if it means that at this stage, "Hey, we need to pump the brakes on this project because there's too much risk involved," or, "There's way too much potential for harm to be caused by the work that we're doing." I think companies are getting more comfortable with that, with that action step, or at least I hope so. Maybe it's just the idealist in me, but I think that it's a place where overall, organizations are doing more critical thinking about all of these topics, and that's super necessary and I hope it continues with time.

Shane McGraw: Great. We'll work in a couple questions from our chat here. Another question from Martin asking, "For tech companies that practice scrum, parentheses (Agile methodologies) how are the research aspects of AI engineering fit into this kind of this kind of work flows?"

Rachel Dzombak: Yeah. I'm happy to start with that one, and then, Matt, you can jump in. You know, I think oftentimes when you are in any sort of process flow, especially, you know, when you're in design-build-test loops, it's great that they're iterative, and to Matt's point, testing is completely necessary and is something that we want to see more of. Testing being done early, being done often, focused on what are the riskiest assumptions embedded in this idea and how do we test for them? And it requires that these teams also step back out of that process from time to time and think about the broader system in which their work exists.

I think oftentimes that's the step that Agile specifically leaves out of popping back up to say, "Wait. Are we--is our work even still aligned with the mission outcome? How has that mission outcome shifted in the past two weeks, two months, six months, and what does that mean for our process?" and I think that there's a hope at times on development sides that somebody else will figure out that problem, that I just need to be told what to do. But I think more and more we're seeing that there's an individual responsibility to pop your head up and ask those broader questions and be comfortable with hearing the answers of, "Hey, things have changed and I need to shift my design and development process at a given time."

And I'll say that, you know, in my world of spanning different engineering disciplines, this is not unique to software. There was a longstanding analogy in the manufacturing space that, you know, the manufacturers would be working on the line and it was up to the design team to throw over the wall the plans, and the manufacturers would just make sure it got built, and over time that field saw that, that breakdown, and recognized there needed to be more integration from, you know, it's really the specificity part that Matt had mentioned earlier of specifying, "What are we trying to do?" with, "How are we going to build it?" and creating a tighter connection between those two things, and so I think more and more we're going to see communication across boundaries, process adjustments, to ensure that we are doing that behavior of popping our heads up, asking why, and ensuring that we're aligned with the mission outcomes, and doing that frequently so that you're not in the space of building towards an outcome that's not going to work out.

Matt Gaston: I might just add a few things. Certainly agree with Rachel's perspective on this, and maybe I'll come down into the details of the processes, right. So in Agile and scrum, those are oftentimes associated with DevOps activities, and I think when organizations are considering AI and specifically machine learning or deep learning capabilities inside of their development process, Agile and scrum, it does introduce new challenges to the overall process, right. Some people talk about data ops and how to create the data sets that drive the capability of these systems, and that is an entire process on itself. How do I accumulate the data I need? How do I know it's the right data? How do I know that it's not riddled with all kinds of challenges and things like bias? How do I version it over time? How do I age off data I don't want?

There's also ML ops, which is sort of a subcomponent of a larger DevOps cycle if you're building a system that includes machine learning, and sometimes model training can take days or weeks, which has an impact on your development cycle time, and so there are interesting ways that I think these AI technologies are pressing on the processes that exist within organizations and will require them to adapt and evolve, but I think, to Rachel's earlier points, these incremental, iterative approaches where we experiment with things and try them out and make sure we're asking the right questions, again, in an iterative fashion, are really, really important.

Rachel Dzombak: And the last thing I'll add to that, just because it's something that has come up a lot in our work, is also that this is the maintenance of these systems is a critical part that we often find organizations are underestimating what maintenance looks like, and especially what does it mean when there's context shift, when there's data shifts? How do you track those things? It's both a--you know, the architecture is evolving at the same time the context is evolving, and that's a difference from traditional software development. So it's a place that there needs to be thought organizationally about, "How are you guiding these processes, guiding the behaviors you want to see across teams?" and I think we're going to continue to see evolu--there's all sorts of combinations of words right now, of DevOps, DevPsych Ops, DevPsych F-Ops, all sorts of permutations, and I think we're going to keep seeing new processes evolve as organizations try them out and figure out how do they support this, these development processes over time?

Shane McGraw: Okay. We got a question form Dallas in the chat. Matt, we'll start with you on this one. "Are you aware of any projects, any--are there any projects that work for poverty alleviation or pensions through AI day trading?"

Matt Gaston: I'm not aware of anything specifically in--on that problem, but I often think if you can imagine it, there's probably someone out there working on something like that. So I don't have anything off the top of my head, but searching around, you'll probably find something similar, and there's a path to follow there.

Shane McGraw: Okay. And then we'll go back to John. John sent in another question earlier asking, "AI will impact on software design--will have impact on software design, development and evolution. Now what ways do you see the role of the software engineer changing as a result?" We can start with you on this one, Matt.

Matt Gaston: Well, there's two interesting sides to this question. There's sort of the, "What does the software engineer do about AI components in their systems?" but there's also, "How does AI help and enable the software engineer?" So the first one, I'll leave it with an easy answer, which is software engineers most likely, given the proliferation of these types of technologies and the commoditization of machine learning capabilities and other AI techniques. Software engineers are, I think, necessarily going to be forced to learn about them, understand how to use these types of tools and technologies in their process.

But maybe the more interesting side for me, and an interesting question that the Software Engineering Institute is doing a lot of research on, is, "How can AI--" oh, "How does AI impact the role of the software engineer and how can it really help or augment what software engineers do?" And so people out there might be familiar with a new capability called Codex. It was jointly developed by OpenAI and Microsoft, and it's one of these very large, so-called transformer machine learning models, right. I won't get into the details of how they work, but they essentially read lots and lots and lots of data. Lots and lots and lots of text, and in the case

of Codex, all of--I think all of the open source code that's available on GitHub, and what Codex can do is from a natural language description generate small snippets of code, and it's really interesting to watch the demo. You can search for this out there and watch OpenAI put on a demo of Codex. By the way, I'm not endorsing a product. I just think it's an interesting use case out there in the world. But you can add, essentially, as a software developer, what you might put as a comment, and then Codex will automatically fill in the code that's there, and it's not perfect by any means today, but it's pretty powerful. The demo includes the creation of a web server and a website. It includes the creation of a game.

Yesterday I heard someone talk about, you know, "I could describe--write me some code that shows snow falling on a black background," or whatever it is, and Codex is capable of creating that automatically, and so I think even in software engineering, right. It's sort of software engineer's building capability that then software engineers can use. There are real interesting questions about human-machine interaction here and how software developers can rely on but maybe not totally trust or hundred percent trust these capabilities, and really interesting system design concepts and code design concepts that are at this intersection of what can be autogenerated, is that easier or harder for then a software developer to go and fix and analyze and test and integrate into other capabilities?

And so there are amazing AI capabilities out there today that are already working on sort of automatic generation of code, right. Today it's small scale, you know, 5, 10, 20 lines of code, but I think there's potential for these things to go much, much further.

Shane McGraw: Rachel, anything to add there?

Rachel Dzombak: Just that I think it speaks to other conversations we've had earlier in this, that the role is going to change and it's why that learning is needed and that curiosity is needed about how things are changing, and I think with any change, we as humans, we get scared, and we worry that this is going to be detrimental for us, it's going to change all organizations and

certainly that we're going to see roles eliminated. But we're also going to see a lot of new types of roles being created. You know, the tools that Matt is talking about, that's effectively lowering the barrier for who can become a software engineer, and that then allows software engineers to ask interesting questions to expand their skills, to apply them to new areas.

I think that there's certainly two ways to look at things. I tend to be more of an optimist, but a realistic optimist at times, but I think it's a part of paying attention and just being willing and open to how these systems will change, how that will affect roles, and I think we're going to see organizations grapple with that over the coming years and I, for one, am excited to see how it emerges.

Shane McGraw: Great. We're down to about five minutes left, so I'm going to combine two questions from our chat, let you guys give a response to that, then we'll go to closing thoughts for each of you before we wrap up, and then last, I'll just mention lots of great questions in chat. If we didn't get to your questions in the chat or that came in via email, I'll make sure we get them to Matt and Rachel to--and there's maybe--we have other ways to respond post-event.

But the questions we're going to combine are from Annie and Philip. Annie asks, "Is there any effort at CMU underway to establish a degree or certification for AI engineering?" and then Philip's question was, "What is the path to become an AI engineer? What skills do we need to have?" So we can--can we start with you on that one, Rachel?

Rachel Dzombak: Yeah. I love this question. So there is already a degree at CMU or a couple degrees that are in this flavor. So there's one that is actually at CMU's Africa campus in Rwanda, Kigali, Rwanda, and so that program is a master's in Engineering of AI Systems. It is in its first iteration right now, and so it is about really tying this core curriculum and engineering systems, engineering AI systems, with domain knowledge of what does it mean to apply AI in agriculture, in manufacturing, and how do you balance context knowledge with this kind of deep technical knowledge?

There are plans in the works for other programs, especially at multiple levels. At the undergraduate level, at the graduate level, as well as for executive education. It's a place that we're definitely interested in helping to evolve and are working with campus right now to think about what those options look like. So that's a long way of saying stay tuned, and things are already happening at the same time.

In terms of skillsets, you know, I think there's a lot of debate around that right now, so this is not an SEI viewpoint, but it's mine. I think that being able to be, you know, have this technical grounding, have a strong foundation across the AI stack that Matt described earlier, is going to be critical, as well as being able to hold the complexity of the problem spaces. I think more and more that's the skill that's going to be needed.

You know, as an educator, my least favorite question is when students come to me and say, "Tell me exactly what I need to do to get an A on this test," and that mindset is not going to fly as we think about how to implement these systems. Instead, there's a lot of uncertainty, a lot of unknowns. Being willing, able to wade into that complexity, to break it down, understand what are the component parts, what does that mean for me in my design decisions? That is going to be key, and it really comes down to this social-technical fluency that I talked about earlier. I think more and more as the tools continue to evolve, the technical barriers are lowered, we're going to see an increasing importance of that, that fluency.

Shane McGraw: Matt, anything to add there, or your closing part? Yeah.

Matt Gaston: I'll do both, Shane. I'll add that there is sort of interesting thinking along a narrow path of how to become the AI engineer, and I think Rachel talked a little bit about that, but I think there's also an interesting phenomena that I'm observing, which is there's a great opportunity to add AI education into the core curriculum across a broad variety of disciplines, of majors, of

courses of study. Much like, you know, 10, 15 years ago, we--a lot of people started talking, or maybe 20 years ago, started talking about computational thinking with the rise of software systems and the influence of software in the world. I think we're seeing the same thing with AI now, right. No matter what you're studying, it's likely to be of interest, to understand how AI works from a science and theory perspective, but then also how to practically apply AI in whatever domain you might be working in. Civil engineering, medical, healthcare, public policy. All of these places have opportunities for these technologies to have a very positive effect, and that brings us back to where Rachel started with a diversity of perspectives is critically important to get this right. Lots of different people with different experiences and different training need to be a part of this.

My final thoughts, Shane. Obviously, we're really excited about AI engineering here at the Software Engineering Institute. We're excited that the narrative is shifting, not just around the excitement around AI itself and what AI can do, but how to build these systems in a responsible and reliable way, and our approach is we want to be a part of it. We want to be a part of creating an AI engineering discipline and an AI engineering movement, but of course we can't do it alone. We don't pretend to have all the answers. We're very much taking a community-based approach, and so we look forward to engaging folks on this webinar today in any number of ways into the future, whether it's education or understanding what things have failed or worked in your particular application space, and any number of ways. So just wanted to emphasize community-based approach. We're going to be doing a lot of things like this in a lot of other ways to engage with that broader community, and excited to see people participating.

Shane McGraw: Great. Rachel, final word for you?

Rachel Dzombak: Yeah. Just for me, one on--just can't help myself. On the pathway to become an ML engineer, I'll just say that no one has this solved yet. We did a really interesting study talking to different organizations about how they were hiring for AI engineers, ML engineers, and there's a lot of open questions, and a lot of it comes down to how do we train internal people to develop these skills and hold them in parallel with kind of the work they were doing previously? So if anybody has more questions on that, happy to have a follow-up on it.

I want to echo everything Matt said about our community-based approach. We feel very strongly that we can't do this alone, and so if there are things you want to contribute, resources you know of, tools you use successfully, case studies that would be worth us to know, positive or negative, please do reach out. I know Shane put a couple of ways in the chat to connect with us. We'd love to hear from you and are continuing to think about how we intentionally move forward and bring a diversity of partners into this effort in the same spirit of having many perspectives on what does it mean to actually engineer these systems and how do we do them in a way that is responsible?

Shane McGraw: Great, Rachel, Matt, great discussion today. Thank you very much for sharing your expertise, and lastly--

Rachel Dzombak: Thank you, Shane.

Shane McGraw: Yes. And lastly, we'd like to thank each and every one of you for attending today. Just a quick note, the CEI is a strategic research partner this year for the 2021 AI World Government. The show takes place October 18th and 19th, and this two-day forum educates federal agency leaders on proven strategies and tactics to deploy AI cognitive technologies. For more information, see www.aiworldgov.com. Upon exiting today, please hit the Like button below and share the archive if you found value today.

Also, you can subscribe to our YouTube channel by clicking on the SEI seal in the lower-right corner of the video window. Lastly, you could join us for our next livestream, which will be Monday, September 27th, and the topic will be "The Future of AI: Scaling AI Through Engineering." Registration information is available on the chat now. It's also on the SEI website, and we will follow up with an email as well.

Any questions from today's event you can send to info@sei.cum.edu. Thanks, everyone. Have a great day.