

SEI Podcasts

Conversations in Software Engineering

We Live in Software: Engineering Societal-Scale Systems

featuring John Robert and Forrest Shull

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Forrest Shull: Hello and welcome to the SEI Podcast Series. My name is [Forrest Shull](#), and I am the lead for defense software acquisition policy research here at the SEI. I am joined today by [John Robert](#), the deputy director of our Software Solutions Division. Very excited, I know John and I have been looking forward to this conversation for a long time given how timely the topic is with all the things we see in the news every day. We are here today to talk about societal-scale systems and the engineering that goes into them. Welcome, John.

John Robert: Thank you so much, Forrest. Happy to be here.

Forrest: Great. One of the things I was really privileged to do recently was I was the co-lead for an IEEE Computer Society technical forum on societal-scale systems, but specifically looking at misinformation, disinformation on social media and digital media platforms. I have really had the chance to see that this topic that we are going to be discussing today really resonates. I think there are a lot of our folks in the computing profession that are just

concerned citizens, feel and see these issues, and understand that these are computing systems that we are talking about. They all want to know what to do about it. But I have also seen ... Obviously, this is a very big business issue. But also, from the point of view of policymakers and government folks, a lot of interest in understanding some of the harmful effects that go along with the good things that these societal-scale systems do and what if anything we might need to do about it. So with that, let's take a step back though and talk about what exactly societal-scale systems are. I wondered if you wanted to take a shot at the definition there, John.

John: Sure, Forrest. Thank you. We have had some conversations as a team on a report that we will talk about in a minute, but the way we decided to define societal-scale systems is that they are fundamentally about the challenges of modeling human behavior at scale when people are integral components to the system. In other words, they are not just people that are using this system and clicking buttons to provide simple inputs, they are really an integral component of the system because the system is designed to really create human response, such as engagement or other things that we will talk about. Companies have developed such systems, for example in social media, to generate engagement and to generate financial revenue, but they can also be used to influence humans in other ways too, which we will talk about.

Some examples include social media platforms, such as all the things you have heard about Facebook and things like that. But there also examples in other kinds of organizations, for example, healthcare, where there is Fitbit and other kinds of things where there is engagement to create human behaviors that are positive behaviors. But really the software is designed to create that engagement for health benefits. Then there are also other examples too. For example, virtual reality, the metaverse kinds of things, and even future topics like smart cities, which is one of the things you have talked about before, how smart cities could be used to create new types of engagement with users and influence human behavior in new ways.

Forrest: Great. A lot of what you mentioned there was important, but the one that always strikes me I guess is that phrase *at scale* that you mentioned. I think some of these things are not new, and I think we have been used to interacting with computers for a long time. But the level of influence they have and the ability to impact things really at a societal scale, I guess, is quite new and I think kind of unexplored territory in some ways.

John: Absolutely. The impact that these systems have on our daily lives is just

really huge now. There are usually hundreds of millions, sometimes a billion users of some of these systems on a daily basis. The kinds of impacts they are having on our society are growing, and some of these systems are engineered to create new kinds of behaviors with people. So understanding how these systems are engineered is a key part of software engineering over the next decade.

Forrest: Great. Yes. Let's explore that part of the engineering there for a moment because I think one of the areas that you and I started talking about these things, first off, was when we were both involved in the [National Agenda study](#) that SEI did, looking at the future of software engineering. I would be curious if you wanted to talk about first, what that study was, and then maybe say a little bit about how these societal-scale systems fit into that larger view that we had on where software engineering as a field might be going.

John: Sure. It was really interesting. The SEI, the Software Engineering Institute, is a federally funded research and development center to advance software engineering for the Department of Defense but also for the software engineering community in general. As part of that mission, we started a study in the very early parts of 2020, so January of 2020, to really look at the next decade, the next 10 years of software engineering research. This is research intended to build the future software systems, to look at where we are today, look at the challenges that are ahead, and what kind of research is needed to create the systems of the future.

As part of that activity, we looked at different types of topics, different types of systems. One of the topics that came about was this area of social media, and we started to realize that these types of societal-scale systems were really a big part of what we were living in at the time. I mean, the pandemic started shortly after that. We as a society and around the world were seeing different kinds of effects of these systems on our society. Our interest is not on any one aspect of the debate about these systems but on how to engineer these systems. What are the new challenges about these systems? As we started to explore that, we realized it was hard to unsee some of those challenges because when you peer under the hood, they really are different kinds of systems.

For example, as we talked about earlier, when these systems are designed, they are created using psychology mechanisms to create a response with users and engage in a new way, not just people providing input, but are really to elicit a psychological response from the people using the systems.

That is in the one-on-one software to one person, individual. Then if you look at that at scale across many, many people, predicting how these systems behave at scale is really a challenge, and that was part of what we were seeing. As we explored that idea, there are some new challenges here, it is a new type of system. And looking ahead, how to engineer these systems, we saw several areas that we will talk about in a minute that needed to be explored, and it was really interesting because we realized just how much these systems are part of our future.

Forrest: One of the things I really appreciated about the study was that it was really community driven, and so it was a great opportunity to interact with a lot of different segments of the computing and software engineering population here in the United States. To your point, that was why [societal-scale systems ended up one of our top six themes listed](#). It was one of these things you just hear over and over again from the folks that we talked to about looking into the future and seeing where the interesting problems and challenges are going to be.

I was going to make the connection. Last year, I had the time of my life actually being the IEEE Computer Society president, and same issue. One of the things I really loved doing was going around and talking to different parts of the computing profession that are represented there in the society. Just the amount of interest and even anxiety I guess I heard around some of these systems, especially the digital platforms. It is in the news every day. I think people worry about the types of information or misinformation that get pushed out about things like vaccines, about the war in Ukraine, about all kinds of issues. And recognizing that underneath these are really, like I said before, computing and software engineering systems that we feel we should, as a profession, have something to do with it. I think there is a lot of interest in these issues there and probably a growing sense of how we engineer these things better as we go.

John: I very much agree, and I think the *aha* moment that we have is thinking about people as really new components of a complex [system of systems](#) and thinking about how, when we engineer these systems, we engineer them to accomplish certain types of properties like safety or security or reliability today. So what are the software engineering activities needed to construct and engineer these systems in ways that are also safe or predictable in some new ways?

Forrest: Yes, I think that parallel with safety and security is an important one actually because I always say underneath it all, I don't think that these are

really natural phenomena although sometimes we treat them as like immutable ways that the systems behave, right?

John: Right.

Forrest: We understand how to engineer systems and we make specific choices to engineer them to be safe and to be secure, and we have tradeoffs that go into all that. I think as we go forward, I think there will be a lot of work in this area. We will understand better the engineering underpinnings that have to be done to understand how to make the systems more fair, less biased, whatever the new set of *-ilities* in a way is going to be.

John: If I could add one more thing, I think one of the other aspects, because you talked about the IEEE community and the input you were seeing and the challenges we were talking about, I think one of the things we have heard from the advisory board of the National Agenda study that I mentioned before, which included people from across industry, Department of Defense, government, [DARPA](#) [Defense Advanced Research Projects Agency], etc., a lot of different perspectives about this new challenge, new area. One of the things that really solidified for us that this is really an area that needs attention, and from a software engineering perspective, is that there are starting to be reports, papers, and even documentaries such as a Netflix documentary, [The Social Dilemma](#), that was created by software developers, people that were creating these systems, people that were maintaining these systems, part of the startup companies, and they were asking questions as well that we felt were very profound in terms of the software engineering challenges ahead.

Forrest: Very similar to that, I think one of the things we see through the Computer Society, oftentimes, in the past, it's always been a technical community obviously. I think a lot of times we're somewhat guilty of treating the technologies that we develop as if they are value-neutral in a way or that they are always going to be used for good. Exploring this issue now through the larger community, I think one of the things that we quickly realize is you bump right up against freedom of speech, some of these other social choices, which is why we had some really good conversations at our technical forums with a congresswoman from the U.S., an MP from Canada, some U.K. government adviser. It strikes me that people are aware of these issues, they are concerned about them, and there are a lot of social choices that have to come into play here. On the tangent, I think it is really important for us as technical folks to be part of those conversations. I think people are hungry to do something and people are willing to do policy in these areas. I would

really like the technical folks to be part of that and have a voice so that they can be well targeted and as effective as possible when they are needed.

John: Very much agree. I think the conversations that you are talking about are also very cross disciplinary, where there is a need to engage with people outside of the software engineering or computer science communities to the other communities of psychology, sociology, even economics and other kinds of disciplines, to really help shape the software engineering techniques and tools needed to solve some of these problems. It is interesting how software engineering as a discipline has needed to extend itself. We used the phrase *think systems, build software*, which means that we have to think at the system level, understand properties and behaviors we need to have to happen and bring that down to the software architecture, software engineering activities that we need to pursue as professional developers. With that in mind, it really reshapes how we think about the boundaries of the system and these kinds of things but also shapes, going forward, what are the new techniques or new areas that we need to have available in our tool set to build these kinds of systems.

Forrest: I think bringing the conversation back around to some of the things we can be doing today, that last point was a great segue into kind of the more constructive part of it. I think one of the key things that is always at the top of the list is that kind of multidisciplinary conversation that you just talked about, John. I think it is always important for us as software engineers to have that domain expertise and be able to work across the discipline boundaries, but I think this is a scenario where it is even more challenging and more important than ever. I think some of the other things I see going on myself is some work on just better tools to monitor the content that goes out there. Not necessarily that anyone is trying to get into the chain of it and kind of disrupt the communication but just to understand what is actually happening.

One of the things I was really surprised but interested to see through the Computer Society work is that there is this whole population of small companies that have sprung up, which are really trying to do the work that some of the larger platforms aren't in terms of monitoring, in terms of testing whether bad content really gets through the filters, and just have a sense of what is out there and be able to report on it because I think that level of insight is already very important for us to do something with it.

John: Agree.

Forrest: Beyond that, I know that the Computer Society is starting some work on standards in the area, trying to look at, *What does it mean technically to be a trusted news source? or How might I do some research, some rating, and some assessment of different news sites in order to say which ones are more trustworthy than others?* I think that is a really hard and challenging thing and honestly, a lot of opinion gets into that, but at the same time, it's a sense of the community doing what it does best, which is coming together to argue and hash out some of these issues and hopefully make some progress even if we can't get the whole thing done.

John: I very much agree with all the points you made, Forrest. I think one of the aspects of this conversation which I found so fascinating is, a lot of times people think that the discipline of software engineering has to change to create a new type of system. Well, these systems have been created, they exist, and there will be new types of systems in this category over the next decade with start-up companies, etc., as people extend them to new areas. What is really needed to advance software engineering isn't how to create these systems, but how to create them better and with a better understanding of what are the tradeoffs that are happening when you build these systems, but also how to meet the society expectations for these kinds of systems in terms of privacy and bias and security and other kinds of concerns. Understanding the new challenges and new qualities that are expected by societies through those policymakers, etc., for these systems is critical, and then having the software engineering techniques and discipline to really build and move in that direction is critical.

Forrest: That is great, and now one of the things that that touches on too is just the need to be conscientious, as engineers but also as citizens, while we use these sites, to have those high expectations about what they should be doing in terms of privacy or what kind of information they push through. And at least be able to make informed choices about what tradeoffs we are willing to make on a day-to-day basis.

John: Yes.

Forrest: This is good. This talks about what we can do today with the systems where they are. One of the things I wanted to circle back on was, I know you've been doing some looking at where things are going, and I think the National Agenda study also looked at where things might be headed in the next 5, 10, 15 years out. I wondered, a [recent blog entry](#), it pulled together some thoughts on five important research areas that are informing these types of systems, and I wondered if you could present a little summary

of that.

Forrest: Sure. Happy to. There are several areas, but as we talked about they are expected to be cross-cutting, cross-disciplinary for all of them working outside of the software engineering and computer science communities with some of the other communities. One of the first areas is new [quality attributes](#) and architectures.

One of the challenges with these types of systems is that these societal-scale systems have new quality attributes that are not well understood. Most software architectures today are created with these tradeoffs in mind, with well-understood quality attributes such as performance or security or reliability. The research area that is needed here over the next 10 years is really a focus on the new quality attributes for societal-scale systems and understanding the relationship of the design decisions as these systems are created with the behaviors that can be expected after they are deployed and being used. Understanding both what are the attributes that need to be engineered but also how to measure those attributes and how to build to create and accomplish those attributes is really critical. New techniques in software engineering are needed to really do that because today, and we have seen in our recent history, that there are a lot of challenges today with these kinds of systems. And trust and privacy and bias are not necessarily new, but the challenge of predicting these kinds of properties at scale like we talked about and really understanding what the consequences are if the systems are built to really focus on the individual interactions and engagement without thinking about the larger-scale problems or properties. It is really a new area that we have to explore and build those new techniques.

The second area, if I could go through that and keep going, developing a theory of sociotechnical knowledge creation. This is a little bit different kind of thinking, but it is the idea that we got from [Jim Herbsleb](#) at Carnegie Mellon, who is a professor leading the teams there in the newly created society computing group [[Software and Societal Systems](#)]. One of the things to think about is that, kind of like a funnel, a lot of the data that we see today in these societal-scale systems are creating new communication mechanisms, new types of information flows. That is wonderful because it connects people around the world in different ways and new ways with many, many benefits. But one of the things that happens is, as the data is going through these new information flows, is that is not processed and then integrated into society's knowledge base in quite the same way. We usually think about reporting in a newspaper or magazine or TV and how

that is processed and sometimes vetted for information. And then, that makes it into society's understanding of our challenge or a problem or a topic, but now there are new pathways. How do those pathways work? What are those pathways and the inputs of those pathways and how they create new outputs in terms of society's knowledge is really important.

I think this is another thing that comes up about our role for the Department of Defense because there are national security challenges if we don't understand this relationship about societal-scale systems and the inputs, and then what does it change in terms of society's understanding it, and knowledge set as an output. We really have to have ways to understand, measure, track these kinds of shifts, and make sure that for the safety and security, things are transparent and not happening totally behind the scenes without understanding what is going on.

Forrest: Yes. That sense of understanding I think is where again, when you do it at scale where so much of the challenges come in. I think it goes far beyond being a technical issue, I guess when you start to look at [AI \[artificial intelligence\]](#) and [ML \[machine learning\]](#) being able to help you identify some of the problematic content that might go out there.

John: Right.

Forrest: There is enough nuance that's hard to do when you're in English. But then, when you think about this kind of global systems, then you have to worry about things in Indonesia and all kinds of populations where we might not have that kind of experience at headquarters, it becomes a real challenge about how to manage that.

John: Excellent point. I agree. When you think about the global challenge in terms of languages, different types of data, like a picture is different than textual data, for example, and video is different from that, and understanding how these different types of data sets are filtered through new communications channels, it is really a wide-open area of research. There have been some publications, and we cite some of this in the report. [Rand Waltzman](#), for example, had done some work on this with DARPA in recent years. But more research is needed. That is the point, is that more research is needed in these topics to build the tools and techniques for engineering these systems going forward.

The third area that we identified in the blog is adherence to policy. As you were mentioning before, policymakers around the world are talking about

societal-scale systems and debating, sometimes even creating new policies or rules to govern societal-scale systems. We identify in the report again that research is needed to really help formulate and validate some data-driven and openly understood techniques for industry, government, and society to really have a technically grounded governance framework, something that is not just sort of a blunt instrument but is something that is really informed by the technology, underlying technologies, and what are the opportunities and risks of those technologies but also really trying to look at, going forward, what are the tradeoffs.

Software engineers should not necessarily be making a really fundamental tradeoff about privacy versus some other quality. They need to be following some general guidance and policies from the government to the point that that is needed, to their company, and then also be transparent about some of what is happening behind the scenes. Again we are focused on not what is the policy but creating a technical framework so software engineers have the skills and the training and the tools they need to create these systems knowing that these policy aspects need to be built in.

Then a fourth area is experimentation and testing. Another big challenge of these systems that became apparent as we talked to folks is that these societal-scale systems really operate at a scale that really creating a test environment is extremely difficult. Replicating a system with so many different types of operators and users with so many diverse opinions and perspectives is almost impossible. So creating a test environment has limits—it can be done, but it has limitations and constraints, and that challenges the way we think about testing a software system, which typically is done in different stages from unit testing to integration testing to operational testing, etc.

So a deeper understanding is needed here about how to test and what does it really mean to test the societal-scale system and how experimentation can be done on operational systems. One of the things that is done in these kinds of systems was called [A/B testing](#), where out of the many millions of users, a slightly different version of the software may be released to different users, and the company or organization that released the software can test whether or not certain small changes of the software result in a desired response that they want from users, like for engagement or clicking on an ad or whatever. That is called A/B testing. What is interesting is that it is a form of software testing, but it does result in changes to the software, but it's done on the operational system. We really need to explore what are the opportunities for testing in an operational environment, but it also includes

what needs to be released to the public in terms of transparency and openness about the kinds of testing that is being performed on a day-to-day usage basis. So testing is another huge area.

Then finally, I would like to talk about software development using these types of systems. One of the niche areas, sort of a small sliver of the large scope of these systems, is when these societal-scale systems are used for developing new software. We think today about software being developed not in a small team but by a distributed team, typically a globally distributed team, and especially with COVID distributed even within a certain city where everybody may have gone into an office, they may be now scattered all around. So that distributed software-development environment is really an interesting opportunity to apply societal-scale systems, where an individual software developer is not just typing in code. We are already seeing today that there is AI-enabled support for the developer that on the integrated development environment provides suggestions about typing as you are typing and uses the context of your development environment to make suggestions for parameters, etc. The next evolution is likely applying the societal-scale influence to where a developer could be incentivized to do certain things. Maybe the developer is incentivized to develop software faster or do more releases in a day. Maybe a developer is incentivized for other concerns like quality, fewer bugs in the software.

We talked about how at some level these things are already happening, if someone thinks about a software developer getting like a year-end bonus for a job well done. At that level, yes, some things are being incentivized today, but the opportunity with these new systems is that there could be far smaller, granular incentives on a weekly or even daily or hourly basis to the software developer to take certain action. The research that is needed here is how effective are these new techniques to really enable the desired results or benefits. Then, also, as we look broadly for the software engineering discipline, what are some of the risks involved with that kind of activity?

Forrest: OK. Now, that example especially about the incentivizing, I think that is a theme that has come up a couple times in this conversation. It is interesting to me because I think sometimes we look at these systems affecting human behavior as like a weird side effect. But no, actually the point of these systems at scale is exactly to influence human behavior. Whether it is like you said, to produce better code, conserve water in a smart city, have traffic move more smoothly, or trust the system enough to use it and make it part of your day-to-day routine. So again, it is one of those things where I think it's important that we have those cross-disciplinary

conversations to understand these types of tools and the psychology behind it, and when it is a good use of it and when it is a little bit something we should be concerned about.

John: Agree.

Forrest: I think that that is also a good segue into the next topic or question I had because I think it is kind of an interesting question about how people can start to work or get involved in this type of research. The societal-scale systems are strange in that they are all around us. We move through them every day, but it is somehow hard to get access to them when they are proprietary, they are commercial, they are kind of behind some kind of protection. I wonder if you have thoughts. I think you have already mentioned some resources that people might use to get started in the area.

John: Sure. I think that the resources are growing. I think one of the things that I found fascinating are documentaries like the Netflix, *The Social Dilemma*, which is for software developers I think an interesting insight. I think you mentioned the IEEE activities. It is great to get involved with professional societies, because professional societies will be a key part of how to work through these questions and develop maybe new solutions. We also see that a lot of universities are starting to have or have recently had a shift toward these societal-scale computing kinds of concerns. At Carnegie Mellon, as I mentioned, Jim Herbsleb changed the name of their whole department, aligning with the society kind of impacts of software writ large. We see at Stanford and other places similar departments that have been in existence, looking more broadly at how software engineering touches all these other disciplines. I think there is opportunity there for anything from coursework to research, and there are probably many others. I think there are some books that you may have...

Forrest: Oh, yes. I have visual aids as a matter of fact. Because I often say, there is a lot of good stuff out there, not all of it written really in deep technical direction for an engineer, but things like [Weapons of Math Destruction](#), which is the best title ever, I decided.

John: Yes.

Forrest: One of the things that book talks about, again, in a very accessible way is the use of these algorithms, how they get embedded into large systems. Then, you have these algorithms that are at least informing, sometimes making decisions, but they are not auditable, so that you don't

know if you get the chance if you are being discriminated against. If there's an algorithm, let's say, that's making decisions about who gets to be alone because who looks risky or who gets to be let out on parole because we have some data about recidivism rates. If those things are not going in your direction, you don't often get the chance to query the algorithm. You don't get the chance to see why that's happening and they get wrapped up in a whole level of prestige and accuracy that might not actually be the case depending on how they were developed. I think it is a message to us as engineers that as we build these systems and encode the algorithms, there are issues about transparency, about fairness, issues like that that we have to consider as we go forward.

The other book I have, which I really liked, again, not even so specific to societal-scale systems per se, the engineering of it, a book called [Invisible Women](#). What this is about is not the algorithm, so much as the data sets that they were trained on. When you have these algorithms making decisions about what kind of medical procedures to recommend, let's say—and they have been trained on data sets in which the female population is underrepresented or not represented at all—it has the possibility of really driving poor recommendations. And again, because we focus on the engineering of the system part of it and maybe not so much the underlying data set, these are ways that pernicious bias and other problems can creep into the systems without us even realizing it sometimes.

I heard you mention the Computer Society work too because the technical forum, we have a list of resources that we have compiled through those conversations that are available for folks. We will make the links to this and the other resources available on our website afterwards. I think there are a lot of those things there that people will find interesting just to hear different perspectives on the conversation and think about how these things might come into our systems.

John: I think you are right. The multiple perspectives that are needed here is really interesting because it is still unclear what all the dimensions are that are needed to really inform an understanding of the problems and create some new solutions. I think everybody is realizing our increasingly dependent nature as a society on the reliability and safety of these systems. I think it will be an incredible area of research and exploration over the next certainly 5 or 10 years.

Forrest: One last thing I thought of too is that I think for especially for educators in software engineering, there is a real opportunity here to do a

better job of bringing in what it means to do ethical engineering. Just from my own history, I will say that it has always been a part of the software engineering curriculum that I have seen that it is always treated as an afterthought or as an add-on.

One of the best practices that I have heard of from some educators was the idea of team teaching a class and bringing in someone with the expertise in ethics. A lot of times, we think that we can understand ethics, and we will teach it to our classes and we give it short shrift because it is a very deep area just like our software engineering one. So bringing in someone who really is that expert and being able to have them be part of the dialogue for the class, that you get to see the give and take of some of these conversations and how different issues might get thought of differently from an engineering and ethical side. I think there is no necessarily one right answer, but understanding that there is a process that you go through to think about these things and worry about how they might come into play is still a very important part of the education that we need to be training people on as we go forward.

John: Absolutely agree because several of the conversations that we had in creating the report is, we thought about possible future scenarios. Again, our intention is not to try and predict the future because that is very difficult in the technology space, but we looked at future scenarios and possible future scenarios to give a sense of the broad set of possible futures that we face. One foundational aspect is that this concept of people as components, of using influence, kind of, concepts as approaches to really shape the user to do certain things or to take certain actions, that simple idea can be extended to so many new areas that we haven't really seen yet. I mentioned the software developer and how to state that, we talked about smart cities and healthcare. One could imagine in financial cases how that might be applied. It can be applied in all kinds of work environments. The ethics concepts that you just highlighted will be critical to help everyone including software engineers understand what are the opportunities for applying these techniques and new systems, but [also] what are the risks and maybe the limits that need to be applied as we look at how this can be used.

Forrest: OK. As you mentioned before, John, I think the SEI has a very specific role in the larger research ecosystem in the U.S. I wonder what is next for us in this area. We talked a lot about things that people can do and awareness and where the larger community or where the larger trends might be going. What is next for SEI here?

John: That is a great question. We are continuing to be very interested in exploring this area. We do have research, fundamental research that we do and applied research we do, and then work we do with customers. We are certainly exploring the space of how societal-scale systems can be used for software-development environments. What does it mean for software developers and social software-development environments to apply this, where is it effective, where is it not effective, doing research to explore that space, that is certainly an area we are going to be tracking and exploring over time. As we look further, the use of AI in these systems is an aspect, one of the other areas that we highlight in the report is what is called [AI engineering](#), that is how do you construct a system that has an AI component in it, and what does that mean for testing and for the whole software lifecycle, we add AI components to it and [MLOps](#) and things like that. As we look to those kinds of concerns, many of the concerns that have to do with AI could also be applied here because there will be new approaches but new risks and concerns. That will also be a theme in our research going forward.

Forrest: Great. Well, thank you so much, John, for the time today. I think this has been a great conversation and hopefully we have managed to convey at least a little bit of the excitement I feel because I think these are really exciting and important challenges to deal with as we go forward.

John: I very much agree. Thank you for the time. I think, as you have said, it is really a key area for software engineering going forward.

Forrest: For our listeners, we will include links in the transcript to the resources that we mentioned during the podcast today, so you will be able to access those and read more about it if you're so inclined. Finally, a reminder to our audience that our podcasts are available on SoundCloud, Stitcher, Apple Podcasts, and Google Podcasts as well as the SEI's YouTube channel. If you like what you see and hear today, give us a thumbs up. Thanks again for joining us.

Thanks for joining us. This episode is available where you download podcasts, including [SoundCloud](#), [Stitcher](#), [TuneIn Radio](#), [Google Podcasts](#), and [Apple Podcasts](#). It is also available on the SEI website at sei.cmu.edu/podcasts and the [SEI's YouTube channel](#). This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu