Is Software Spoiling Us?

Table of Contents

SEI Panel Is Software Spoiling Us?

SEI PANEL

Is Software Spoiling Us?

Jeff Boleng, Moderator
Grace Lewis
Satya Venneti
Eliezer Kanal
Joseph Yankel

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University Notice

Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM17-0943

Carnegie Mellon University Video Notice

**003 Speaker: And hello, from the campus of Carnegie Mellon University in Pittsburgh, Pennsylvania. We welcome you to Virtual SEI.

Virtual SEI's our new streaming platform where you can watch live events or access on-demand videos discussing our cybersecurity and software engineering research and best practices. Our presentation today is a panel discussion on, "Is Software Spoiling Us?"

My name is Shane McGraw, your audience moderator for the presentation, and I'd like to thank you for attending, and we'll make today as interactive as possible, so we will address questions throughout the discussion. You can submit your questions at any time during the

presentation by using the Q&A or Chat tabs on the page interface. As with all our events, we ask that you fill out our survey upon leaving today's event. Your feedback is greatly appreciated. The link to the survey will be in the Chat area soon.

Now I'd like to introduce our panel moderator for today. Jeff Boleng is the Chief Technology Officer, as well as a principal researcher at Carnegie Mellon University Software Engineering Institute. He joined the SEI in 2012, after 21 years of service as an active duty cyber--active duty cyber operations officer in the United States Air Force. Now I'd like to turn it over to Jeff. Jeff, all yours.

Speaker: Thanks, Shane. Today we're going to talk about, "Is Software Spoiling Us?" which is a little bit of a play on what a lot of people think about traditionally when they think about software in the--especially with DoD acquisition and government software, that it's usually always late or over budget.

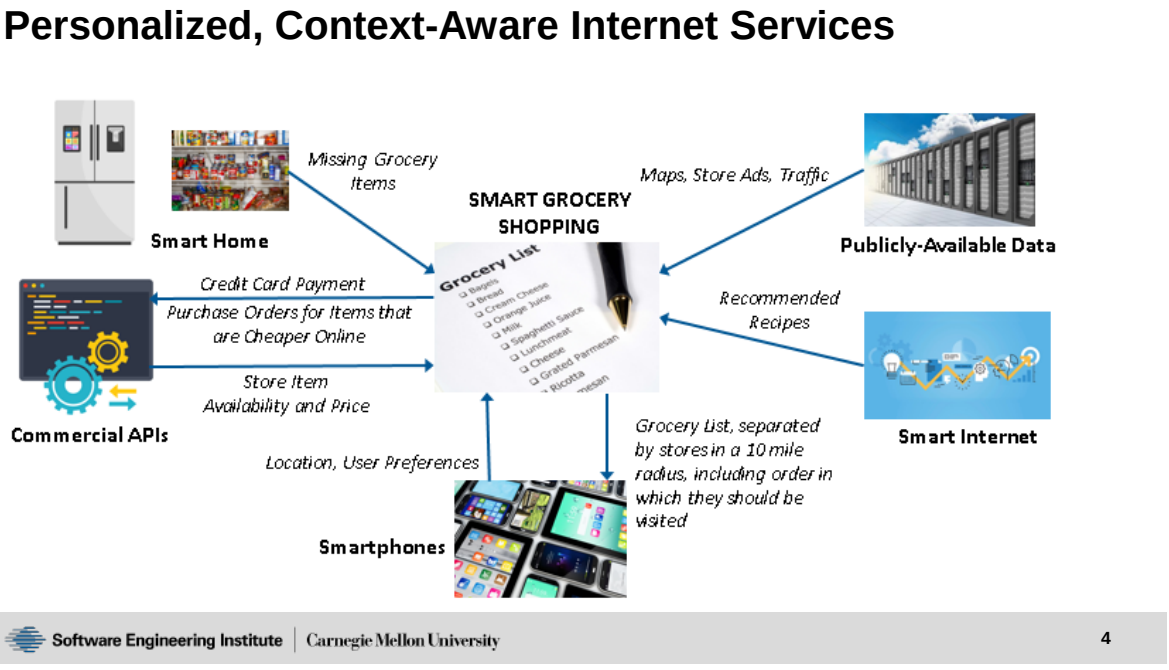
But I think if you look at the successes that the commercial world has, implementing rapid capability development and rapid capability realization through software, it's really remarkable and we want to, going to talk today, about some big successes the commercial world's had and how we might be able to translate that into the ability for government and DoD to develop their software more effectively, more quickly, more securely, more cheaply.

I think there's a bit of a mismatch between how well software's achieving capability, performing for us in our commercial world and civilian lives, and the way that the government and DoD has been able to leverage some of the same things.

So we're going to start today, I'm going to start with Dr. Grace Lewis. She's a Principal Researcher here at SEI. She focuses on a wide variety of things from IoT security to cyber foraging to something we call cloudlets here at SEI.

But Grace, first question, and we're going to go around each of them and give everybody a chance to answer this, about what is something awesome in your daily life, not your DoD or--

Personalized, Context-Aware Internet Services



Error! Not a valid embedded object.
**004 Speaker: Right.

Speaker: --government life, but in your daily life, that, something awesome that software, some awesome capability achieved through software?

Speaker: Right. So to me it's amazing that nowadays we have these personalized context-aware internet services and that they can be composed with things that exist today. So I was just reading something over the weekend and it made me think about this smart grocery shopping, which is something that is perfectly achievable today.

So you can imagine a scenario in which a user has a smartphone and the user says, "Okay. Time to go shopping," right? And the smartphone sends their user preferences to the smart grocery shopping services. It sends its location, and basically what the smart grocery shopping service does is put together a grocery list, and how can it do that? Well, it can use elements from my smartphone, look at my smart pantry, smart refrigerator, and tell me what grocery items I'm missing.

Speaker: Do you have a smart pantry?

Speaker: I'm working on it.

Speaker: Okay.

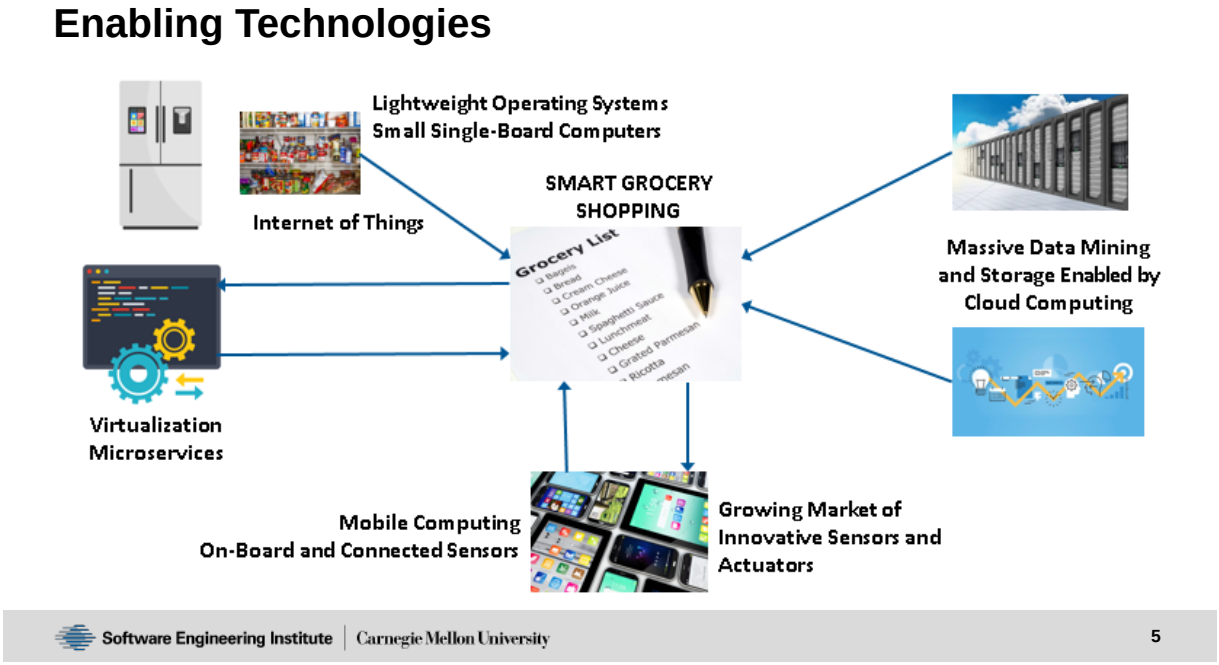
Speaker: That's cool.

Speaker: It can use publicly available data. It can use map data to see where stores are located. It

can go through store ads. It can look at traffic information. It can use commercial APIs. Lots of stores have APIs nowadays and you can use a commercial API to say, "Is this item available, and how much does it cost?" The service on top of that could use what I call the smart internet. I like cooking. Well, actually, I don't, but--I like cooking and it knows that I browse through these recipes, and I mark this recipe as, "This is something that I would like to make this week," and so it can grab ideas and items from that. In addition to that, the smart service can say, "Well, I mean, sure. I know you like actually going shopping to a brick and mortar store, but there might be something cheaper online."

It can look for items online. If they're cheaper it can buy those and pay for it with your credit card, and so at the end of this, this personalized context-aware internet service, what I have on my smartphone is a grocery list. It has a list of stores and it has what I should buy at each store and they're all within 10-mile radius of my house, because that's what I told it to do, and this is not science fiction. It's something that is perfectly, perfectly feasible today. So.

Enabling Technologies



**005 What are the technologies that are enabling that? Well, there are lots of them. So first of all, massive data mining and storage enabled by cloud computing. Lots of data in the cloud. Lots of just data mining capabilities, algorithms, that try to make sense of all the data that's out there. So not only thinking beyond just, you know, through my recipes and see what I like to cook, but beyond that.

Internet of Things. Big thing nowadays, and people are getting very, very creative with the devices they can build. Lightweight operating systems that you can put on very small, single-board computers and you can--all of a sudden you have a smart pantry. Going back to your question.

Speaker: Right.

Speaker: Virtualization, microservices. That's how companies are putting their APIs out there so you can use them in your applications. Mobile devices. I mean, smartphones are getting smarter and smarter. On-board connectors, connected sensors. Things that you can plug in. Weather, to measure water quality, air quality, fitbits, health meters, whatever. So again, these technologies are all available today and being able to build these apps, and like you said, not knowing that I'm using all these services from everywhere, and it's just amazing to me.

Speaker: Yeah. I mean, you truly get penetration with a technology when it becomes ubiquitous and--

Speaker: Right.

Speaker: --and you forget that you're interacting with it.

Speaker: Right.

Speaker: So all right. We're going to go to--thanks, Grace. That was the highlight of dozens of things, of cool things we're leveraging.

Speaker: Right.

Speaker: All right. We're going to move to Satya Venneti. She's a Senior Research Scientist here at SEI. One of her primary areas of focus is machine emotional intelligence or having machines help understand what the state of the human being.

Success Stories – Machine Emotional Intelligence

Machines detect, understand, and respond to human emotions:

Human-Human Teaming

Humanyze's sensor-laden ID badges analyze speech, activity, and stress patterns to enhance human-human interaction, collaboration, and communication in the work place.

Human Machine Teaming

BRAIQ teaches autonomous vehicles how to read the comfort level of passengers and learn to drive the way they prefer, increasing passenger comfort and fostering trust in tech.

Augmenting Human Capabilities

Cogito analyzes speaking patterns and conversational dynamics between call center agents and customers, providing real-time guidance to better engage customers.

Offsetting Human Limitations

SAM helps spot suicidal teens by analyzing their language and social media activity.

Adapted from https://techcrunch.com/2016/12/02/emotionally-intelligent-computers-may-already-have-a-higher-eq-than-you/

**006 So Satya, what--give us some awesomeness that software brings to us.

Speaker: So I'm really excited about machine emotional intelligence, and really it's, I think of it, as a shift from the information age to an actual communication age. So it's actually able to, you know, machines are able to detect, understand and respond to human users' emotions in real-time, and that can actually help in many ways, so I, I kind of listed four here, which is human-human teaming.

So it can actually help humans interact better with humans, and so there's one startup out there called Humanyze, and it came out of MIT, and they have these ID badges which actually have sensors on them, and they are in real-time able to transmit

stress patterns and movement and so on, and so they actually help to understand how employees are engaged and improve processes and so on. So they're actually using it on Wall Street to actually look at, you know, how their bankers are being stressed out. So I think that's amazing.

And then the other thing is human machine teaming. So again, there's the startup called BRAIQ, or B-R-A-I-Q, which is actually, you know, like, they have these self-driving cars and they equip them with sensors and they're actually, you know, actually looking at how people feel when they are in these self-driving cars and actually, you know, looking at how comfortable they are when they're accelerating or braking or so on.

And then the other one is augmenting human capabilities. So again, you know, there's this startup from MIT and it looks at speaking patterns and it's actually able to help call center agents to engage better with their customers. Like, "Hey, you're speaking too fast," or you're actually speaking over each other or you're interrupting each other.

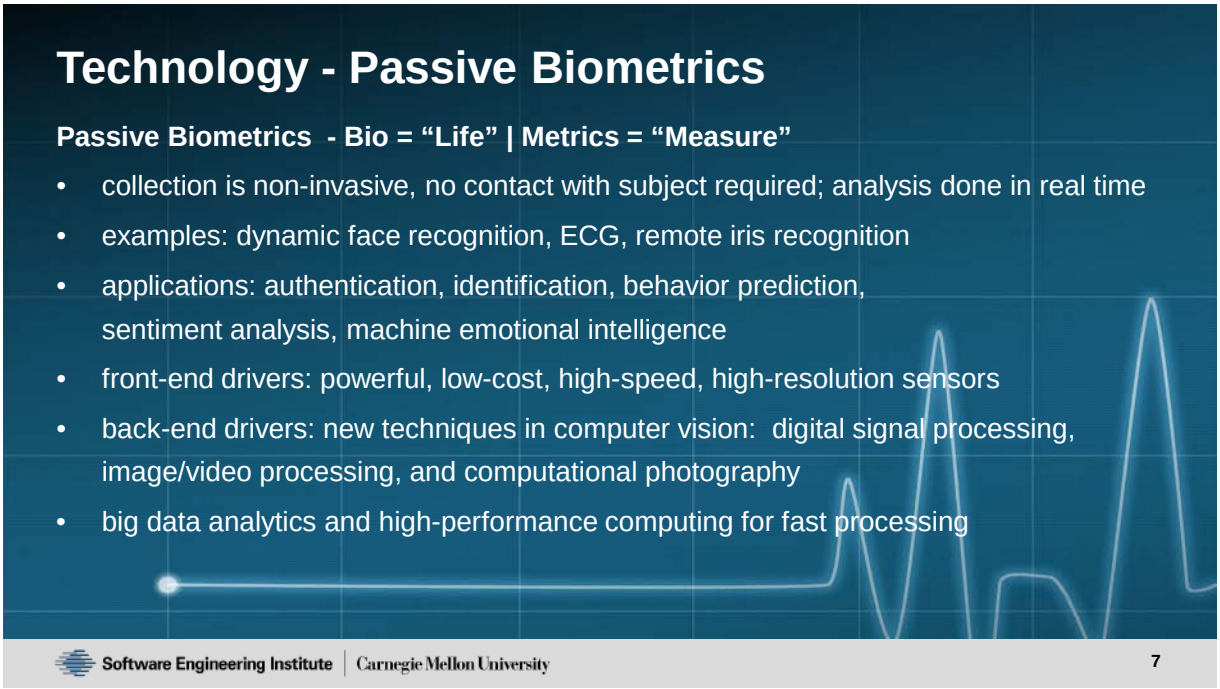
And then the last, and my favorite, is offsetting human limitations. So there's this actual app called SAM, and researchers are actually seeing if teens are suicidal by actually looking at their patterns of social media and so on, and that's a big problem out there today.

Speaker: Mm-hm. Mm-hm.

Speaker: I mean, I think girls, you know, actually teen girl suicide is, I think, at a 40-year high and so this is something that I think, you know, machine emotional intelligence is really helping to actually look at how people interact with each other, machines and so on. I think that's great.

Speaker: So what kind of technologies power all these innovations?

Technology - Passive Biometrics

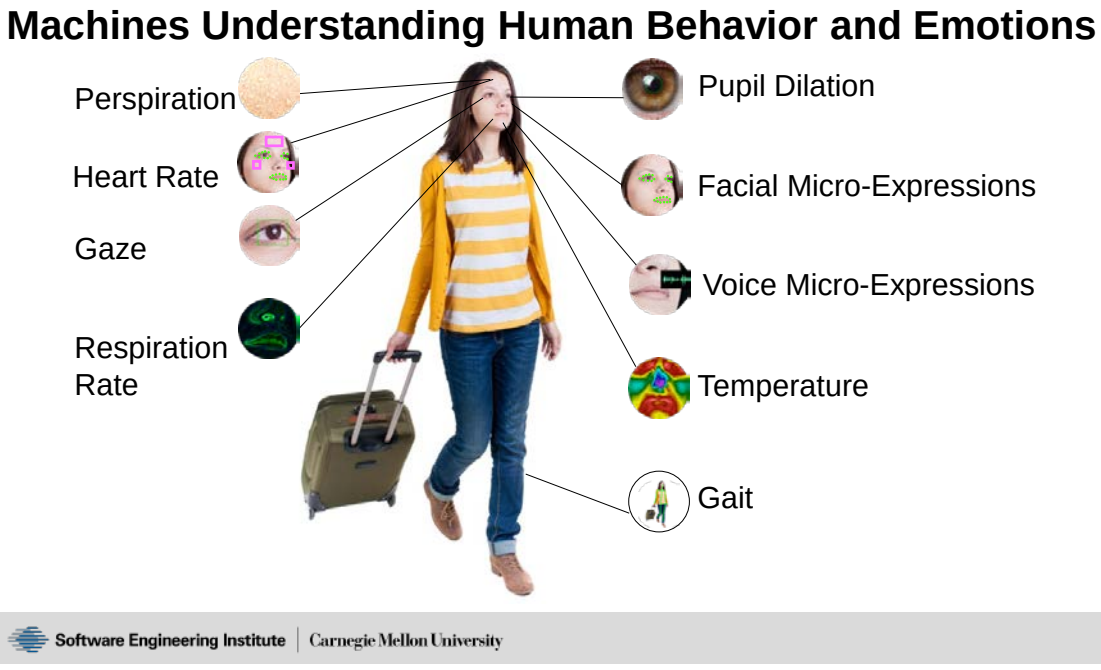


**007 Speaker: So there's something called passive biometrics, and it's this new generation of biometrics where we actually are able to look at people and, you know, actually without contact. We are actually able to see and collect

biometric information in real-time and actually analyze that in real-time, and the drivers are that on the front end we have these low-cost, high-speed, high-resolution sensors that are available, and on the back end we have new technique software, you know, actual signal processing, image and video processing and so on, and of course, big data analytics, that you can take all this data and crunch it in real-time and see how the person's feeling in real-time. So I think this is the main driver for this whole, you know, machine emotional intelligence.

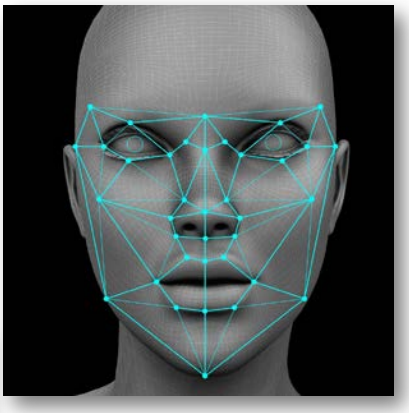
Speaker: Awesome. Thanks. I think it's time. Now we're going to try--we're going to go to Eli Kanal. He's a Tech Manager--

Machines Understanding Human Behavior and Emotions



**008 --In our CERT Cybersecurity Division.

Examples



Software Engineering Institute | Carnegie Mellon University 9

**009 And Eli's built a data science team that's pretty formidable for us here, and he leads that and is one of them too. So same question. We're going to do, we're going to get through all four of us on this question, through all four of you on this question. What's some examples of awesome things software can really, has brought to, society?

Speaker: So there's a couple. One of my favorite ones that I'll start with first is the success of the AlphaGo software in being able to not only play a game that is exceedingly difficult, exceedingly abstract, with a huge number of possible moves, but it's also able to actually behave with what we would refer to in humans as creativity. So we're starting to see the machines not only, you know, the software is not only able to perform a

task it's given, but come up with new ways to do the task and that literally outshine what the humans have come up with before, and that, that coming out of something which we've built, is really pretty impressive. You know, the people who build these almost refer to it as their children, as they're watching their child grow up to do something pretty impressive.

Speaker: Yeah. I read that the win for AlphaGo was a strategy that no human--

Speaker: No human has ever seen.

Speaker: --had seen before. Yeah.

Speaker: I don't know much about Go, but I remember reading them discussing it and they say that the middle of the game the software placed a piece in somewhere which didn't make any sense, so that any of the people who are watching it, you know, they're wondering if it was a bug, and as they finished watching the game unfold, this enormous, beautiful strategy came out, and you hear the masters of the game who've been playing this and are international champions were describing it as incredibly elegant, and they were expressing the sort of amazing admiration for the software the same way they would be of one of their human peers. So it's a pretty impressive achievement.

Speaker: That's a--so that topic's enough for a whole 'nother webcast,

but it brings up all the questions like, "Is it repeatable now?" you know.

Speaker: So it's interesting. So let's get to that. So one of the other areas where I think we've seen some extreme impressive advances are in the areas of self-driving cars. People don't tend to think of it, but a self-driving car is much more difficult than just put a train on the track and, you know, put a robot on the speed pedal. There's an enormous amount of freedom. You know, the car could drive on the curb. There's enormous amount of social mores to have take into account. If a person's coming down the street, a human driver will slow down a little bit to indicate they've seen the person, even if they don't need to slow down. That kind of behavior is very difficult to train into a machine. So the ability to do that has really, you know, it shows kind of how far software has come that we're able to train the machine not only to behave and perform the task but to perform the task in a way that it can actually interact with the human peers, where the robot is going.

You asked before about some of the technologies that underlie this. There's an absolute ton of them in the field of machine learning, but just to focus on one for a second, there's something called a recommender engine which can look at other types of, what behaviors, have you done before? And based on what you did before, you know, we were talking earlier about shopping. Based on

what you've done before, it looks like you might like this thing.

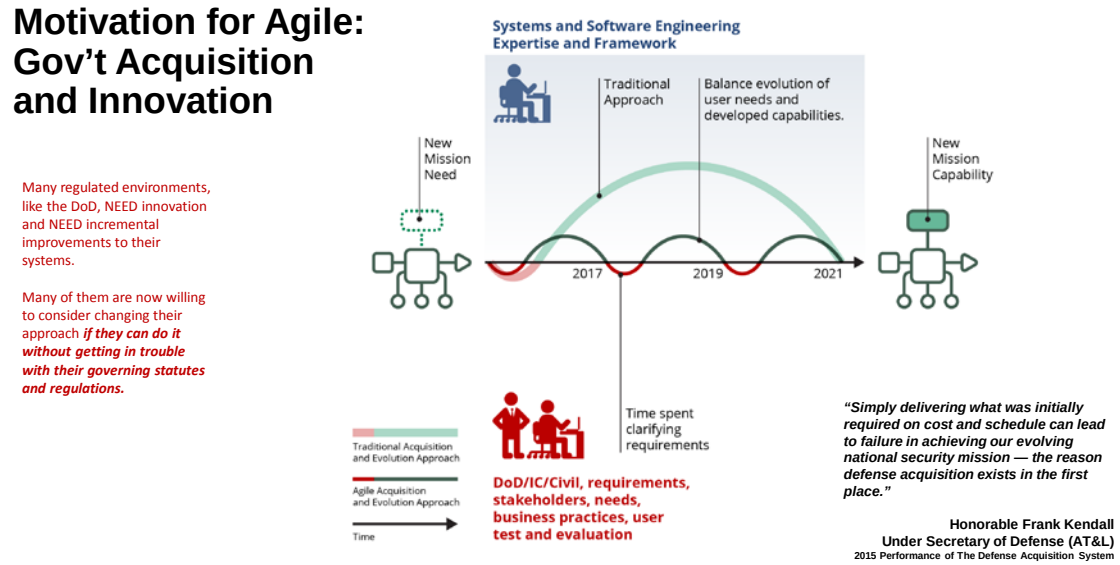
Speaker: Right.

Speaker: You've never bought that before, and in fact, this may not even be in an area where you're familiar, but other people who have liked what you've liked have liked this, and these advances and the way they've implemented them have really kind of brought some of this stuff into the next level.

Speaker: I love recommender engine, because I hate shopping and I want to go to the website and I want to say, "I want something like this and you know all the other crap I bought. Give me the thing you know I'm going to be happy with," and it does, and I love recommender engines, actually.

Okay. So last time for this question. We're going to go to Joe Yankel. He's a Senior Software Engineer in our, also our Cybersecurity Division, but he really focuses on secure DevOps. So Joe, I think Joe's going to tell us about how DevOps saved healthcare. No, I'm kidding.

Motivation for Agile: Gov't Acquisition and Innovation



**010 Speaker: Well, actually, I recently did read a, I guess, an article or I watched a presentation on a fellow who was brought in as a contractor. He worked for Google at the time. His name's Mikey. I forget his last name. Sorry, Mikey. He's now the head of a new department established in the White House, Digital Services, and he was brought in. Healthcare is a big, big thing, right. Health reform hasn't been really attempted in decades. Administration does, and they're on the line to get this up, and what happens?

Well, four million people try to register the first day, the site doesn't work. They wanted to save it. They brought in a team of people. They came in and looked at the situation and said, "Wow, it--well, let's look at

the monitoring." "Well, we don't have monitoring. We can't tell you what services are up, what's down. We have no idea how to do this," and there his thought was, "Well, let's just supply--let's supply the basic techniques of DevOps. Let's get in there, let's get everybody together in a room and work through the problems iteratively, one step at a time, until it works." Three months later there's eight million people enrolled. Big success.

We do focus on DevOps. It's unique for every company. There's not really a standard path. There's not a tool that works for everybody. You have to come in, assess the situation, which could be unique for every organization, every business, and come up with a plan that works and it involves quite a bit of communication. Communication's key. It's something we've been practicing here with our customers and the DoD and it's been working. It allows for us to practice. We practice Agile.

Agile hasn't been typically done in software acquisition. You spend a lot of time with requirements. You propose a solution that will deliver a product in three years. Lot of things change in three years. Security requirements, technology. The old way of doing things doesn't allow--

Speaker: Two generations of Processors have.

Speaker: Right.

Speaker: Yeah. And so we want to be iterative in Agile and DevOps. We need to restructure how we acquire software, how we commission the building of software, to allow for new technologies, new ideas, new requirements, and so our focus has been on Agile and DevOps and it does allow this to happen.

Speaker: Yeah. So Joe introduces a--oh, actually, I'm going to not let myself off the hook. I got to talk about something that I think is awesome that software achieved. So I have two examples, actually. But one example a lot of people that know me have heard me tell this, is I'm not fortunate enough to own a Tesla. I wish I was, but in 2014, there were tens of thousands of Teslas made that had adaptive cruise control. Pretty cool technology if you ever been in a car with adaptive cruise control. It's a really neat innovation that keeps you safe and keeps you at the right/same distance behind somebody.

Two years later, while those cars were tucked in in their garages, nestled, you know, silently for the night, an over-the-air software update happened that gave them bio, like, almost complete self-driving capability. At least on the highway, anyway. It's not full self-driving, but it was two years after those cars were produced, a software update allowed a significant increase in their capability, and I'm just floored by that that we're able to do that now. No technician came to the house, no

new sensors were added, no new hardware, no new--nobody bent more metal. So that's one example.

The other example, I think, is a little bit quirky, I guess. In 2013, there was a Vietnamese kid named Dong Nguyen, in three days one weekend he wrote a piece of software that six months later was earning him $50,000 a day, and that was Flappy Birds. Right. If we all remember the Flappy Birds--

Speaker: Yeah. Yeah, yeah, yeah.

Speaker: --application. Yeah, three days. He wrote that thing in three days, and so one of the, the lesson there for me, is what enabled him to write that, to write that software so quickly that could achieve such significant impact? It really, you could--it didn't cure cancer, but it had significant global impact. There was probably an awful lot of productivity lost because of Flappy Birds at work, right.

So I'm going to save my answer for what that, what enabled that, until after we do another round of questions, and then I'll come back and hopefully it'll be a little bit of a teaser. Let me check in with Shane real quick. Any online questions?

Speaker: So we've been having some connection issues, so we've been pushing people to Adobe Connect. So we're just going to keep the conversation going--

Speaker: Okay.

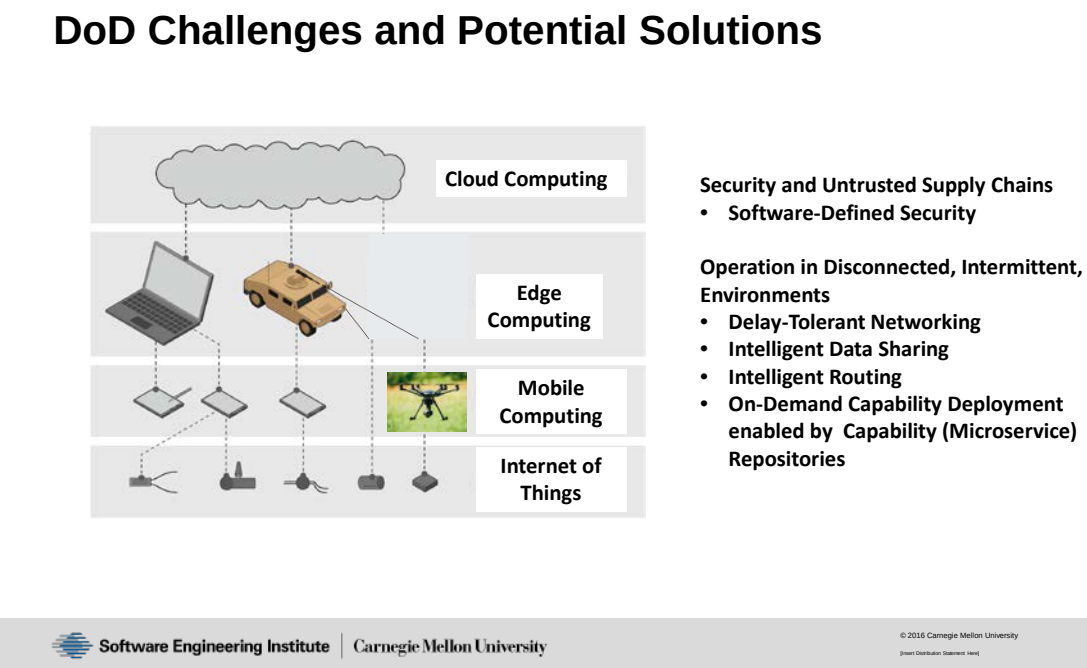
Speaker: --for now, Jeff.

Speaker: All right. So Satya--I'm going to--no. I'm going to come back to Grace actually, sorry. We're going to then now talk about--talked about software awesomeness. What are the enabling technologies? I want to get into start talking about how some of those technologies can help DoD and government and maybe even why we haven't seen them helping as much.

Speaker: Right.

Speaker: What are some of those barriers and things we can do about that?

DoD Challenges and Potential Solutions



**011 And we're just kind of free-

form from here. We'll go through one more round of questions and sort of free-form some ideas on--

Speaker: Right.

Speaker: --how we might be able to better leverage these advances.

Speaker: Yeah. So all the technologies that I talked about, I see them combined in improved situational awareness and I think that's something that the DoD could leverage all these things together.

So if we talk about improved situational awareness using cloud computing, using the Internet of Things, using mobile computing, but on top of that we introduce this concept of edge computing where you can imagine that you have these little clouds, baby clouds, whatever you want to call them. But the idea is that you push pieces of a cloud onto these computing nodes that are in proximity of mobile devices, of IoT devices, and then now you've brought the cloud to them.

They have capabilities that they can use for improved situational awareness. They have data sets that they can use. They have platforms on which IoT devices and mobile devices can load data and that data gets sent to the cloud at some point for processing.

So the idea is this continuum from cloud computing to edge computing to mobile computing to Internet of

Things that really improve situational awareness because you're bringing computing to the data, which is something that has been talked about before instead of bringing data to computer.

Speaker: Right.

Speaker: Anyway, but some challenges that I understand are important challenges, and real challenges for DoD is, one, is security. That's always been a big challenge when it comes to cloud computing, and when it comes to IoT devices, a big concern is untrusted supply chains. Because, I mean, IoT devices, everybody's building one now, right. There are tons of them, and I think we, the DoD, should be able to leverage those.

Speaker: Well, by and large, most IoT devices need to reach back to a cloud.

Speaker: Right. Right.

Speaker: For some purposes, whether they're compute--

Speaker: Right.

Speaker: Compute poor or whatever they are, but--

Speaker: Right.

Speaker: So DoD doesn't typically like to give small sensors to people that continuously call home, right.

Speaker: Right, right, right. Yeah. But to address some of these concerns, I mean, a technology that is--you--I guess it's still emerging, but software-defined networking, software-defined security. Being able to adapt security postures as things change in the network and the threat model. I think that that topic is going, or that technology, is going to bring a little bit or should bring a little bit of relief to DoD, especially when it comes to improving situational awareness at the edge.

Another challenge that DoD deals with, especially at the edge, that my smart grocery list doesn't deal with, is operation in DIL environment, disconnected, intermittent, limited. Of course, if you want to have data flowing back and forth, it's not always possible, right. So technologies that we can leverage there are, you know, delay-tolerant network. I know, Jeff, that's a topic that you like a lot.

Speaker: Yeah, yeah.

Speaker: But--

Speaker: And named-data networking too.

Speaker: That's right. That's right.

Speaker: I'm bullish on named-data networking now.

Speaker: That's right. I'll let you introduce that one. But delay-tolerant networking, to be able to

deal with periods of poor connections, and intelligent data sharing where we know what to share, when to share and to whom, so we're not just spreading data all over the place.

Intelligent routing. Being able to use each other to be able to send data from point A to point B and being able to leverage maybe data that is available at the network level to do that, and also, like I said before, being able, if you're going to be able to push the cloud, you know, to the edge, you have to have ways to package those capabilities. Whether it's data sets, whether it's some very intense, you know, machine learning algorithm, whatever it is. But being able to package those and being able to imagine having a repository of containers, a repository of virtual machines that have these capabilities and being able to push them out to the edge, whether it's on demand, because I'm at the edge and I need this capability, or whether it's pre-provision.

If you know you're going to be in a situation where you're not going to have connectivity, being able to go to a central repository and say, "I want to put this on my edge note and being able to take it out there. So the ways in which I think the DoD could leverage a lot of the technologies that I talked about would be improve situational awareness, especially at the edge.

Speaker: Thanks. Satya, actually, talking about packaging the components--

Speaker: Yes.

Speaker: --in smart ways. Satya and I traveled to a NATO exercise in Romania this last summer and the translation engine for Google Translate on your phone, once you downloaded the language pack, for Romanian, it operated without network connectivity.

Speaker: Right.

Speaker: Because we were sort of out in the middle of nowhere anyway, but it was awesome. You can hold the phone up to any Romanian text and it would translate as best it could to English.

Speaker: Right.

Speaker: And we used it like crazy.

Speaker: Right. So you can have, you can imagine, the edge notes having all that information on there. Absolutely.

Speaker: Right. Yeah. I've heard Cisco, I think, refer to that as fog computing. Everybody's got their own name, right? Cloudlets. Fog.

Speaker: Yeah. There's dew computing now, so--

Speaker: Is there dew computing?

Speaker: There is dew computing.

Speaker: It's a little wetter than fog computing.

Speaker: Exactly. But yeah, everybody--but in the end it's more or less the same.

Speaker: Yeah.

Speaker: It's being able to be able to be closer to the edge.

Speaker: Yeah, push the computer to the edge.

Speaker: Right.

Speaker: Yeah. I've read that, like, the instrumentation on motor jet engines creates terabytes of data per flight hour.

Speaker: Imagine that.

Speaker: And you can't, you just can't move that data to the cloud or to the compute. You've got to move the compute to the data.

Speaker: Right.

Speaker: Yeah.

Speaker: So okay. We're going to go to Satya then and talk again--

Machine Emotional Intelligence in the DoD

Image courtesy of Wikipedia

Image courtesy of USAF

**012 --About the technologies that have enabled some of the civilian and commercial applications and how we might then start to use some of those in DoD or benefit from some of those in DoD.

Speaker: Yeah. So I think there's already lots of human machine teaming going on in the DoD. So on the left-hand side that's called BigDog and it's a robotic pack mule, and on the right-hand side you have a loyal wingman, so it's this--it's a swarm of flying agents which are autonomous, but there's an F-35 in the middle, which has just one human in it. So it's already using a lot of human machine teaming, and what we really need is for humans to trust machines but also machines to trust humans, and that's how you

build a rapport between humans and machines.

So if machines should trust humans, machines should understand them. Be able to predict what they're thinking or, you know, and that's why we need machine emotional intelligence in the DoD. But I think some of the factors that are actually coming on the way is, of course, the big moral issues and the ethics issues. So if the machine can, you know, collect all this data about me without my sort of knowledge, how will that data be used and am I always under surveillance and why sometimes creeps in and, you know, if you use machine learning and deep learning, sometimes it's a black box. You don't know what's going on, and there's those instances where, you know, there was--it was trained only on a certain race of people and when they then used black people it just, the whole, the whole algorithm just failed.

So there's this big mistrust about, you know, ethics and, you know, how it's going to be used, but also I think, I think that the DoD needs to become more humancentric in their approach and thinking, so it's, you know, like, it's always been about the tech. But now we need to think about humans and how humans and machines interact together. So we really need to bring in the human element and so I think there's this thing called the Third Offset Strategy where it talks about human machine teaming and how humans are very...

Speaker: One of the main tenants, yeah. One of the major tenants of Third Offset.

Speaker: Exactly. And it's like the one secret sauce. Like, it's, you know, our people, our, you know, our secret, and nobody can steal them from us, and so it's important to actually recognize that humans are important part of that equation and make humans and machines work together better.

Speaker: Yeah. I heard one of the city Army leaders say at a conference that as we do pursue human machine teaming, if we replace humans with machines, that one plus one has to be greater than two, which sounds like a little buzz word. But really what the point is we don't want one-for-one capability replacement with--we take a soldier out of harm's way and put a machine there to team with somebody else. That thing, that combination, needs to be far more capable, far more lethal, than the two soldiers were before that, so...

Speaker: Exactly. I think humans and machines together can achieve greater things than just a human or just a machine because each of them sort of augments the other one and helps them.

Speaker: What's the--you've got that example about the chess playing.

Speaker: Yeah. So, you know, so this whole thing when, you know,

Garry Kasparov, who was the reigning chess champion, and this was 20 years ago, he was beaten by Deep Blue, and it was like--it was actually like, you know, when actually people started getting very worried about machine replacing humans, and so at that time what happened was people just had this big mistrust about machines and then eight years later there was this whole new, you know, it was a freestyle chess tournament that was arranged by Garry Kasparov where we could have teams of just humans or just machines or hybrids of humans and machines, and guess who won that tournament?

So, you know, I mean, it wasn't like a machine, it wasn't a human, it was a team of relatively weak humans and weak machines but they had a really good process, and that's really, I think, a very powerful result because it shows that, you know, that the actual sum can be actually greater than each of the parts together and that's very important, I think, for us to understand, that it's not actually machines replacing us, it's us working with machines to achieve greater things.

I like to think about intelligence augmentation and not A.I., which is artificial intelligence. So we actually use, actually use machines to augment our intelligence, and that's how I like to think about that.

Speaker: Yeah. I'm a terminal optimist.

Speaker: Me too.

Speaker: And so I hate looking at the down-side. That's why the title for today's panel was, "Is Software Spoiling Us?" not, "Why is Software so Terrible?" right?

Speaker: Yeah.

Speaker: So yeah. And I too like the intelligence augmentation. I've already outsourced my memory to my cell phone. Since I don't--

Speaker: I think a lot of us have.

Speaker: Well, my daughter gives me a hard time. She's like, "Dad, what's my cell phone number?" I'm like, "I don't know. I just click on your face, and then it calls you."

Speaker: So I don't know how to spell anymore, because there's autocorrect all the time, and I don't think that's a bad thing.

Speaker: Right.

Speaker: I actually think it's a good thing because now I can think about bigger and better things. I can be more creative about other things.

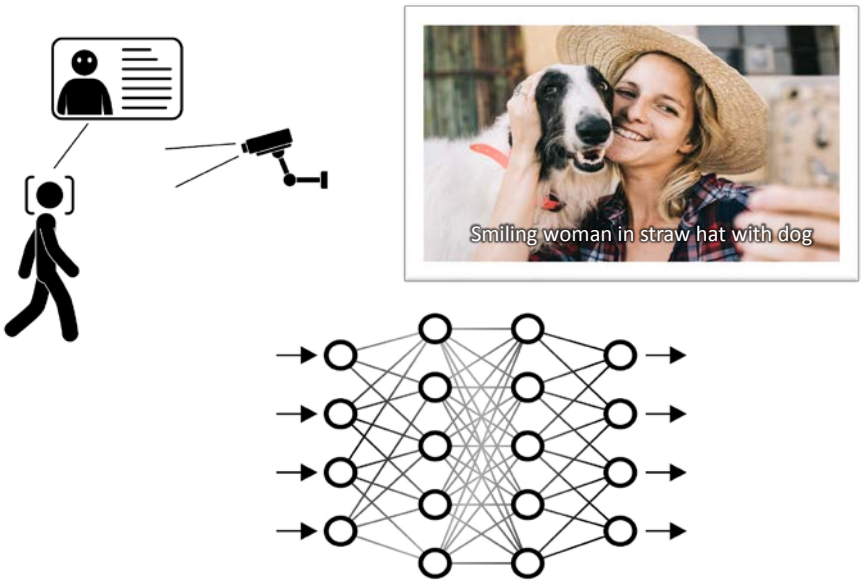
Speaker: Yeah.

Speaker: So...

Speaker: All right. Eli, we're going to jump to you. Same thing. So some of the enabling technologies. How might we better help, enable

and empower DoD and the government to help benefit from some of those things?

Images



**013 Speaker: So it'd be a lot of what we've been saying earlier so far has already kind of touched a lot on the different A.I. aspects. I mean just to highlight maybe one or two other ones. Image recognition. You know, lot of satellite imagery that's coming in.

So the DoD could definitely use some automatic understanding of what's in the image. I have a slide here. It's interesting because Google, and as well--I don't want to highlight Google too much. Many of the other players in this A.I. field have gone beyond simple image recognition, that they can now identify that this is a smiling woman with a straw hat with her

dog, and there's contextual information. They could actually start extracting where she's sitting. There's a lot to be done with images, so that sort of--that says a really obvious relevance, I think, to the DoD.

There's different areas. Kind of ties back to what we were saying earlier about chess and Go and cars. There's a lot to be said about automated decision-making, and having an algorithm that can take in all the input situational awareness, as was being referred to before, if the algorithm knows everything, it's going to remember it a lot better. Sure. We can give it priorities, but it can definitely help with the decision-making process. So there's an awful lot to be said for that.

The main problems that they've been, have kind of stymied getting this all into the DoD, first of all, these all rely on huge troves of data. We have that at the DoD. In fact, they have far more data than they can handle, but the problem is it's frequently siloed and segmented. Necessarily so. So, you know, this group can't see that data and this group can't see that data, because this data's highly sensitive and we really have to be careful what it is that we want to put together.

There's a growing recognition everywhere, including in the government, that data is a liability. If I have data, that means the bad guys

can get that data. So the only way to not let them get it is to not have it.

Speaker: Yeah. But really, the flip-side is the most true, right? Data is not a liability. Data's like--data's the lifeblood of--

Speaker: Data's--

Speaker: --of modern--

Speaker: --simultaneously--

Speaker: --corporations, modern capabilities.

Speaker: Oh, yeah. Oh, yeah.

Speaker: I mean, and that actually cultivating and curating that data properly is--

Speaker: Data is--

Speaker: Yeah.

Speaker: --discipline that we need to get better at.

Speaker: Yeah. It's simultaneously the fuel of all this magic, and it's the source of so many problems, and figuring out how to properly manage that is a risk balance that a lot of areas, including the government, is still trying to figure out. In particular, when you have--if I have a certain amount of data, what's now being recognized, the government has always known--I like those, example, I'm thinking one of the Tom Clancy novels, that some analyst figured out

that there was an attack happening because he saw that there was an upsurge in the amount of pizza ordered.

Speaker: Sure. Yeah, yeah, yeah, yeah, yeah.

Speaker: Yeah.

Speaker: What do they call that? The Domino's Effect or something? Yeah.

Speaker: Yeah.

Speaker: Exactly. So that's a very well-known example. But when you have a lot of data, there's a lot of small clues and the metadata leakage becomes very big. You were talking before about downloading the language pack to figure that out. Well, all of a sudden, if you're using Google's algorithm, Google knows that you just downloaded the language pack.

Speaker: Right. And they know where I am.

Speaker: If you're on the cloud and you're not using it on your computer, they may actually know what you're interpreting. So there's a lot of risks and the government is still trying to figure out how to get past this. A lot of what we were saying about the fog and the edge, that's starting to solve that.

Speaker: Right.

Speaker: And it's finally starting to make its way into DoD systems.

Speaker: Yeah. Somebody at this conference a long time ago told me applications age like fish, and data ages like fine wine. Meaning that the data's the important thing to persist.

Speaker: Yes.

Speaker: And applications come and go. Analytics come and go.

Speaker: Yes.

Speaker: But if we preserve that data.

Speaker: Yeah, interesting. You know, was McNeely, the former CEO of Sun. Way back in late '80s, early '90s, he basically said, "Privacy's dead. Get over it." You know, that maybe we should just--and I think the younger generation has a very different concept of--

Speaker: Absolutely.

Speaker: So there's--

Speaker: Concept of privacy and, I mean, I look at my daughters and the things that they're willing to Snapchat and Instagram about and--yeah.

Speaker: There was a very interesting talk at a conference a year plus change ago where they showed that a Amazon--not Amazon, excuse me. An Android phone with

zero permissions. So you have granted it the ability to do nothing. Just by you walking around can figure out what city you're in because it'll map the path of your walking using the gyroscope to known, maps of known cities--

Speaker: Wow.

Speaker: --and figure out where you are. No permissions whatsoever, and it can do locations.

Speaker: Even tracking--

Speaker: It has location tracking. Going to find out where your house is, which stores you shop.

Speaker: Right, right, right.

Speaker: It's a scary amount of metadata leakage.

Speaker: Yeah. So you just got to learn to ignore it and then--

Speaker: Just got to learn to be careful.

Speaker: --just hope for the best. Let's just--

Speaker: No privacy.

Speaker: Let just trust everybody again. No.

All right. So Joe, let's go to you. Same question on this round. The technologies that, like, helped with healthcare got to go, because you're

the--you jumped in and actually really did highlight an area where modern software development practices really did help the government achieve capability more effectively than they could. Which, like you said, led to the formation of defense digital service and some of those other things that are going on now. Pull in on that thread a little more. What are some examples of how we might be able to better take those technologies to benefit the government and DoD?

Speaker: We just need to start to use them. We see our big successes. You've talked about Amazons and Googles. One thing they do well is they put out a new product. Multiple times a day, right?

Speaker: Sure. Yeah. Continuous release, yeah.

Speaker: It's pretty continuous integration. Continuous deployment. It's quite unbelievable. It is unique often to web presences, right, where you might--

Speaker: Yes, yeah. Mm-hm.

Speaker: --"I need 20 new updates." But what happens is incredible. You know, I have a developer somewhere committing code and I have enough automation in the system to know that I've--my security's good, my code works. Everything's been done so I know this can go live.

Speaker: Yeah.

Speaker: Massive amounts of automated tasks.

Speaker: Massive amounts.

Speaker: Which we really have ignored a lot in DoD and government.

Speaker: We ignore a lot of security. Security is mostly a concern after something bad happens.

Speaker: Right. It's a way to place blame.

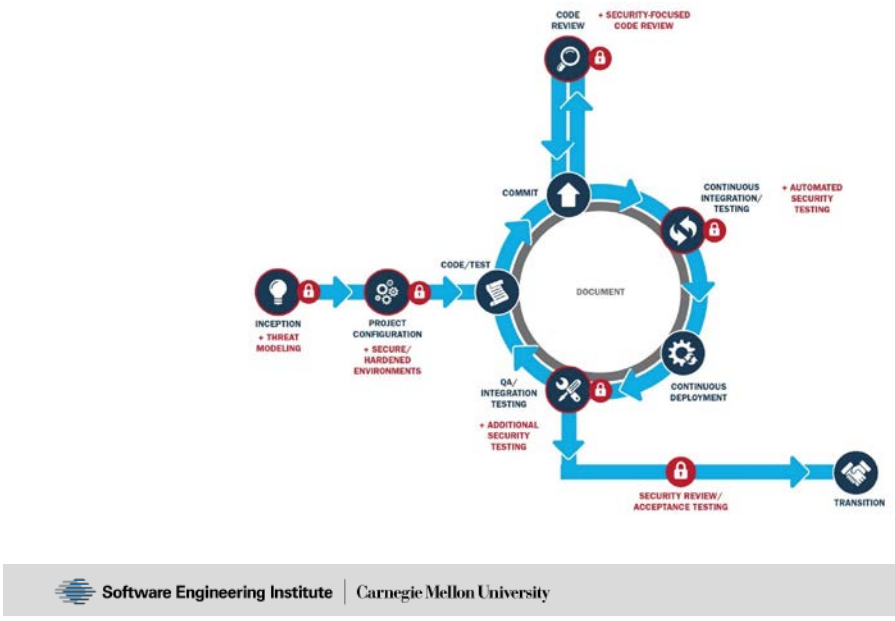
Speaker: Right. Right. We want to spend money once we've lost--

Speaker: Attribution. Yeah, we--

Speaker: --lost some big money.

Speaker: We want attribution. We don't--yeah.

Speaker: What we want to do is we want to bring all the stakeholders in a project together. We want to change acquisition. We want to say, "If this is what we're trying to build or this is the product we want...."



**014 This is software, I need everyone here. I need some business folks. I need security professionals to talk about the implications at the beginning of my stages, or I need testers to talk about what has to happen. I need the operations team.

Operations, IT operations, and the operators often, which are the end users, to have some say in this process from the very beginning and to be included throughout the development, and so this is just a new thought process for DoD, which has been relatively waterfall, relatively contract-based, which doesn't include--well, let's think about it. We have three years to get a project done. I usually don't allocate a person from the government. For three years.

He's going to receive this in three years and then take a look at it, and then say, "Wow. This--I can't get it to work. I don't know anything about this. Let me read about it. I'll get back to you in about six months and let you know what we think about this." Well, that's got to stop. We need them to be involved early. I need provide people and personnel in the beginning to receive this.

One thing we want to push is early prototype, and so what I want is the Hello World version of an application delivered in a production-like environment very early. That way everyone has a chance to look at this. To talk about security implications, to talk about patching, to talk about supply chain. We can do threat modeling very early. We can catch things. We have a better chance of catching things. We have monitoring. You do all the things that you tack on later, and in lieu of bad requirement gathering or old requirement gathering, this allows me to be iterative and change things. You often don't know what you want until you see it.

Speaker: Right.

Speaker: Right. That's commonly true is the--yeah, and the first release is never the best, never right on target, and if you iterate--the other thing I find is if you iterate more often, you're taking your risk in smaller bites, right, rather than multi-years of risk and then at the end of that if you haven't achieved the goal.

You can't scrap the system because you're already--

Speaker: Right.

Speaker: --hundreds of millions or billions of dollars into it, so now you got to figure out how to fix it.

Speaker: So it's like the MVP.

Speaker: Yeah, the minimum viable product.

Speaker: Yeah.

Speaker: It's exactly that, but so here's my trick question though, which what does Hello World look like for a weapons system? I mean, so granted, DoD and government have an awful lot of business IT systems. We have to run ourselves like a business like businesses do. We have an awful lot of even software systems that aren't weapons systems necessarily. Command and control systems, intelligence support.

Speaker: Sure.

Speaker: A lot of that, and a lot of that's very appropriate to do, a lot more rapid application development, right. Build a DevOps team that's got user representatives and operations and security altogether working on it piece by piece. What does Hello World for a weapons system look like?

Speaker: So I think if you think of it as a big, you know, as a big

monolithic structure it's harder, but if you break it up into modules then you can have Hello World for each of those modules, right, and I think that's the secret of doing it, in my opinion.

Speaker: Yes. I think we haven't asked. We haven't asked for them to deliver us a weapon system that allows me to test the small changes. If I know in the future I'm going to make a change to the system and I don't--can't get my hands on it--

Speaker: Right.

Speaker: --it's a million dollars. I need you to provide me a system or a simulated environment that guarantees me a high-level success. Right. Something that's very close to that system that I can keep and I can work on. I think they probably have it in hand.

Speaker: Yeah.

Speaker: They've probably built it throughout their own companies. You just, as an acquirer of software, need to say, "I would like to be able to test this also in my house where I'm going to operate it on."

Speaker: Something Satya brought up is a trend of the government owning the technical baseline. Or you take that system-level design and we don't buy--maybe in the future we don't buy a ship or we don't buy a fighter aircraft. Maybe we buy some avionic software and an

engine and an air frame and we--and we manage that baseline and we get and we start to integrate that. I mean, do you think government's--

Speaker: It's kind of interesting, because--

Speaker: --capable?

Speaker: --in the hobbyist realm, there's a very popular piece of hardware/software called the Arduino.

Speaker: Yep.

Speaker: Right.

Speaker: And when someone wants to build an Arduino, for those who aren't familiar with Arduino, it's basically LEGOs.

Speaker: Right.

Speaker: And you simply say, "All right. I want to build something that's going to sort my LEGOs for me automatically." So I'm going to need to buy a conveyor belt. I'm going to need to buy a camera. I'm going to need to buy a moving arm. I'm going to need to buy buckets that could possibly move, and then you consolidate that whole thing into a system, but you can test each piece individually.

So does my image recognition work? Can I check the color? You know, does my conveyor belt move properly and actually sort it? And when you

piece the whole system together such that instead of having one large system of a capability, you have many small systems, each of which has its own capability. You end up getting a much more manageable system.

Speaker: Right. And you can do the MVP up front for each of those modules, right up front.

Speaker: Mm-hm.

Speaker: Right. But I think the important thing is they all need to interfere as well together and they need to have, you know, like, good APIs and--

Speaker: Right.

Speaker: --and likely--

Speaker: Microservices we're talking, almost.

Speaker: Exactly. And lightweight protocols to talk between them and--

Speaker: But with microservices, you know, it's a different architecture. We need to think about now I need to be able to monitor all these services separately.

Speaker: Right.

Speaker: So it just heightens that. I need things in place now to look at each service. Is it doing its job? But that architecture does work. It

allows small unit testing. It allows me to substitute another piece.

Speaker: Yes, exactly.

Speaker: Replace it very easily.

Speaker: Or maybe go to a different vendor and not have such terrible vendor lock-in. So one of the things that sort of breaks my heart is--not sort of--breaks my heart as a taxpayer is we bought the F-22 years ago, designed, what, 20-some years ago. Great, great aircraft. Most capable fifth-gen fighter in the world right now. We paid a lot for the software, the 20-plus million lines of code in the airplane. Same company's building the F-35. I think we rebought all that software. I don't know how much of the lower-level reuse there was, right. Breaks my heart to see. Northrop Grumman won the contract on the B-21, the Long Range Strike Bomber. Probably going to rebuy all that same software again. So I wish we had the Arduino or what are the--what's the LEGO--

Speaker: Mindstorm.

Speaker: Mindstorm.

Speaker: Is it Mindstorm? Yeah.

Speaker: Mm-hm.

Speaker: The Mindstorms for weapon systems. That'd be great.

Speaker: Or as you were talking earlier, you know, the concept of

over-the-air updates. As you start to separate out what each system does, the sensors are in place. You know, let's keep it with the sensors are in place. So I have this hardware. I can update it, and now just tack on the Arduino capability. You know, this, the F-22. I need to have a different kind of radar on it. Snap. You know, that's the kind of capability you want to get to.

Speaker: So interestingly, the unmanned aerial systems community has started to maybe innovate a little more than the manned aircraft community because they see their platform as more of a bus with payloads.

Speaker: Right.

Speaker: And so they standardize the pods that go on some of the UAS's now, more and more are standardized on open systems interfaces.

Speaker: Right.

Speaker: So that multiple vendors can build different capabilities to be carried on those platforms, and that's a lot more modular of an architecture.

Speaker: Right.

Speaker: So...

Speaker: And given that you're talking about software, you can also simulate hardware with software so

that things don't have to be all built at the same time. Right.

Speaker: Right, right.

Speaker: And that way you start defining also the interfaces to that hardware when it becomes available.

Speaker: Right.

Speaker: Yeah. A lot of--

Speaker: So that's another way. Right.

Speaker: --model-based engineering and doing virtual integration--

Speaker: Right.

Speaker: --is a great capability though.

Speaker: Right.

Speaker: Something Joe touched on really sparked an interest to me. Oh. The monitoring of. We wouldn't think of building a physical system without putting ways to monitor it. I mean, even our cars. We've got sensors in them that tell us when we need to have our oil changed, when our brakes are starting to wear too much. All kinds of things. Just the odometer in the car is a type of monitor, right?

Speaker: Right.

Speaker: So more and more, we're advocating that to build that

monitoring into the software components so that, one, we can continuously keep track of whether it's operating correctly, if it's meeting its insurance cases, right, from a testing standpoint and a security standpoint, and it's going to be especially important as some of these systems do continuous learning, making sure that we can specify our rules of engagement for that software, and the continuous monitoring monitors that and then there's--yeah. They're just continuously monitoring the correctness of the software. I think we've got to get towards--oh, and one of the things, if we're continuously monitoring it, what are we generating?

Speaker: Data.

Speaker: Tons of data, right?

Speaker: A lot. A lot of ML.

Speaker: Tons of ML data, and sometimes, if we know the sensor it came from and the conditions associated with the system, it might be almost self-labeled, right?

Speaker: Mm-hm, right.

Speaker: Because the biggest hurdle with machine learning training data is labeling, and if we instrument our systems properly and we curate that data, we might have self-labeled training data, which that's kind of a nirvana, right?

Speaker: Very useful.

Speaker: So getting into some of that, I mean, some of the stuff that we're doing here at SEI, and actually jointly with CMU, at the same time, relates to exactly that sort of stuff. We have, one of the projects, is, it's coming out of CMU, but it relates to jet engine monitoring, and as the data comes out, not only are you able to monitor the systems and just see what do the metrics show, you know, how are the different parts wearing, but you could then start applying artificial intelligence or machine learning algorithms to say, "Based on how this one wore, this one's going to wear differently," or more importantly, "Based on how the system as a whole is wearing..." you might want to take it out for maintenance early. Because if you take it out now you can just tweak. If you take it out in two weeks, it might be a more expensive fix, and just tracking that--

Speaker: Or if you wait two weeks and a day, it might be a catastrophic problem.

Speaker: Exactly. Exactly. And as you start to watch the data that's coming through from the monitoring, you can really see this, and it's pretty cool what we've been able to do here with looking over those logs.

Speaker: Another comment on monitoring.

Speaker: So this might be relevant-- it was a--

Speaker: You got one?

Speaker: --comment from Jack, asked, it was relevant to this slide anyway, so I'll read it. It was, "It's not just about security but operational availability. This cannot be assessed during DevOps but only after deployment and integration." Is that something you can comment to?

Speaker: Yeah. That--does somebody want to comment that? Operational availability, absolutely, is-see, security's only a piece of operational availability. Actually making the system available and useful during operations is the key.

Speaker: Right.

Speaker: It is. We--well, the feedback's important from that. We do have to learn from these things, so it's critical that operations--we hope it works at that point, but if it doesn't, we need to know. We need to learn from that.

Speaker: I think we should also build, I think, resilience into systems, right. So you should have full tolerance built into it. So if something fails, something else takes over.

Speaker: And it comes to that usability and security are frequently posed as tradeoffs.

Speaker: Right.

Speaker: Mm-hm.

Speaker: So if you're going to have a system which is fully secure--I remember one system, I'm not going to name a company, but a friend of mine was working somewhere. The IT team had proposed a very secure system, which he referred to as a data gel. Why was it secure? Because data could go in. There was no way to carry it back out.

Speaker: Back out. Yeah. Yeah.

Speaker: And it was great. The data was there.

Speaker: Right.

Speaker: But you couldn't use it, and it was a situation where they had completely forgotten about, you know, making sure that operationally it has value. So it's--

Speaker: Sure.

Speaker: --always a tradeoff to make sure you're taken care of. I guess one of the big issues which you'd want to address is a concept which I think we've also done some work here with, is that of threat modeling.

Speaker: Right.

Speaker: So there's a number of ways you can get into how to appropriately set up what is security for your system. But one of the most important things to do is just recognize not everyone's threat landscape is the same. Personally, I

have taken my own steps to make sure that my personal information is secure, and that involves, you know, two-factor authentication, using whatever kinds of password manager I have.

Speaker: Strong encryption at rest.

Speaker: Strong encryption at rest possible. Well, I tell you, for myself, I actually don't do that on my computer because the additional cost required to login and access the data, I didn't feel it was worth it. My information I didn't feel was that valuable. Someone else might, you know, you might come to a different discussion, and that's the whole point--

Speaker: I've already said my data's not valuable, so--

Speaker: That's--well, that's the whole point of the threat modeling.

Speaker: Right.

Speaker: Yeah.

Speaker: So someone who's in the DoD environment, you know, and they go through threat modeling and their data's being seeked for by people in nation-states of different types, they can, they'll figure out what their landscape is and who they're trying to protect against.

Speaker: Yeah.

Speaker: Well, we want to shift operations left. Right. Want those guys involved early--

Speaker: Oh, yeah.

Speaker: --so we can do threat modeling. So they can get what they're actually asking for.

Speaker: Mm-hm.

Speaker: It's really important.

Speaker: Yeah.

Speaker: That communication.

Speaker: Something Satya brought up was we didn't actually script much of this. We just had some candidate questions, but it's ticking off all my great little pet peeves here. This whole thing about fail-safe design, and maybe we can spend some time talking about that, is I always use the physical world examples of really good fail-safe design, like the Westinghouse air brake for trains. It's still the technology, it was patented over a hundred years ago, still the technology, predominate technology, that trains use because it's got a fail--a bunch of fail-safe design factors to it, and Westinghouse being a Pittsburgh company is relevant to where we're at now too.

The other one is the Otis Elevator patent that keeps the tension, the weight of the car, keeps tension on the cable, and if that weight drops,

the safety mechanisms automatically engage. So those types of fail-safe design have served us really well in the physical world. What are some ways that we can do that with software? I mean, so watchdog timers, right, hardware watchdogs are one, right?

Speaker: Mm-hm.

Speaker: Continuously monitor. If you've got an interrupt loop and if you don't continue to tickle the watchdog, it reboots the system or something like that. What are some other ways, some other software fail-safe pieces?

Speaker: Trying to start and free-thinking through it, I mean, when you're designing--

Speaker: I sprung that. I sprung this on all of you. I did.

Speaker: Yeah, no. That's fine. So trying to design a fail-safe mechanism, the one that I'm familiar with--the elevator one is great. There's another one I'm familiar with where it's one of the circular saws. The way they design it, the risk is that someone would hurt themself. So the way they design it, if you're not actively depressing the power button, there's a clamp on the circular saw, and as soon as you let go of it, no matter what the scenario is, that saw stops and it's a clamp, actually electromagnetic clamp. So it's going to stop real quick.

So to that extent, and you're trying to design software, you know, what is the nature of the algorithm? One way which I can think of right now, as you called it, encryption at rest. That's a perfect way to say, "When I'm not touching my system, no one should be able to read anything. As soon as I need it, then I can call it back up and pick it through."

Another interesting one, which we actually see a lot now in banking apps, on the different mobile devices, is that when you switch screens. So if you have a banking app in front of you with all sorts of financial information, as soon as you go to look at other screens it covers the whole picture with just a image. So that way if someone's thumbing through all the active apps on your phone--

Speaker: Oh, yeah, yeah. They don't know what it is.

Speaker: --They don't see what information is present. They don't see your--they don't see the numbers.

Speaker: Tricky.

Speaker: And it's a fail-safe way of making sure that the information that you potentially could leak simply isn't present, unless you're actively doing a task.

Speaker: Right.

Speaker: Yeah.

Speaker: So there's a--

Speaker: You bring up a good point, right. You get to describe what is the failure case?

Speaker: Right.

Speaker: Which is different for every piece of software.

Speaker: Right.

Speaker: Right. And for example, this isn't necessarily fail-safe, but I don't know if you guys have heard of Netflix and their Chaos Monkey and now the SimianArmy, it's a great piece of software, and now it's open-source. Because basically what Netflix has said from the beginning is, "We are going to make our systems fail live, because we want to see what happens and we want to be able to prepare better for what it is," and I think that's a--now you hear about defensive program in computing but that's basically what they do. They prioritize--it's almost like threat modeling, but in the opposite where, like, "What is the most important thing for us?" For them it's streaming. So if anything fails, it doesn't matter what fails. We will keep streaming. If you can't recommend, who cares? Well, you care, right. Because you said you love recommender systems.

Speaker: I do love recommender systems. Yeah. So actually, same thing on Netflix.

Speaker: Right.

Speaker: But the biggest thing for us was when I finally got new, different profiles for different family members.

Speaker: Right, right.

Speaker: Because actually there was a joke on this on the radio. Was like Strawberry Shortcake and Narco or something, right, side by side?

Speaker: Yeah.

Speaker: Because I watched one of my daughters watch the other, you know what I mean, so...

Speaker: Yeah. But that's a, I mean, it's getting, it's moving into the left. Like you were saying, it's just building systems in a way that we know how they're going to fail and we're going to try do our best that if they fail, try to gracefully degrade and do our best. It's a great example.

Speaker: And just in case someone's--just in case someone's not familiar with what Chaos Monkey and SimianArmy--

Speaker: Yeah.

Speaker: --Which is, it is one of the most genius--

Speaker: It's amazing. It's genius.

Speaker: --Concoctions.

Speaker: Amazon decided that they're fail-safe, that thing they want to prevent, is down time. So there should never be down time. So how do you make sure there's never down time? Well, the way you make sure there's no down time is cause your own down time--

Speaker: Right.

Speaker: --And make sure your fail-safe mechanisms are in place.

Speaker: Right. That's exactly--

Speaker: So they have this automated set of scripts which randomly kill servers, knock off services, destroy entire data fields.

Speaker: That's Chaos Monkey, right?

Speaker: That's Chaos Monkey.

Speaker: Monkey.

Speaker: Which is now SimianArmy.

Speaker: Correct.

Speaker: And it goes around and by making sure that they're hurting themselves, they're making sure they're robust enough to prevent it. It's--

Speaker: So that's a strong lesson I think for DoD is for years and years and even still today I'd say the vast majority of the time, the one thing that makes our military I think so

effective is our doctrine and our training and our exercises. We really do train well the way we fight, but we rarely do it in the computer security and cybersecurity realm. We do have, we have a lot of security exercises now where we do offensive and defensive cyber against each other.

Speaker: Sure.

Speaker: But in a no-kidding kinetic exercise, we rarely let them take the computers down, and we should. We need to rehearse it that way and practice that way and do the whole-- some more Chaos Monkey.

Speaker: Well, I remember something called Hack the Pentagon, I think it--

Speaker: Mm-hm.

Speaker: It is. I think those kind of things are great where you actually open up to, you know--

Speaker: The bug.

Speaker: --Everybody else.

Speaker: Yeah.

Speaker: Yeah, exactly, and then actually try to find things that are wrong with your system.

Speaker: Right.

Speaker: So I think that's a, I think, a step in the right direction. We should be doing that.

Speaker: It's interesting because we do, humans, have been doing this Chaos Monkey thing for a while.

Speaker: Right.

Speaker: We call it a fire drill.

Speaker: Right.

Speaker: Right. Fire drill is saying, "Let's just pretend there's a fire, and everyone, can you get outside in time?" We do this actually in the IT field not so uncommonly where the IT group will send out e-mails that look like phishing that aren't really and then to see who clicks on it and then when they click on it they get education.

Speaker: Right.

Speaker: And those kind of services are essentially Chaos Monkey for humans.

Speaker: Right.

Speaker: And that's worthwhile and we need to continue doing that. But I think we were saying before, it's definitely worthwhile, you know, break your own, break your stuff, to make sure you survive.

Speaker: Right.

Speaker: Yeah, mm-hm.

Speaker: Well, we were having--to speak of failing in streaming, we were having major audio issues for

the first 15, 20 minutes of this event, but hopefully the recording we have captured is going to do that, but--

Speaker: Do we need to go repeat everything we just said?

Speaker: We kept streaming. We kept fighting through, but there was some issue that hopefully the archive that we're going to present to people will keep, will get the full glimpse of the conversation, because it was really, really good.

Question from the audience though said, "How would--" or, "What would the war fighter in the battlefield gain from more of the software, the good software stuff you've been talking about? What's the benefit they can expect?"

Speaker: Yeah.

Speaker: I--

Speaker: I think there's a lot of ways that go with that.

Speaker: Think it could be limitless.

Speaker: There--

Speaker: I would throw out unprecedented situational awareness.

Speaker: Absolutely.

Speaker: The proliferation of senses on the modern battlefield is phenomenal, so--

Speaker: Being able to process data right there. Not having to wait for it to go to the cloud and come back, but just doing it there. I mean, I think that's great.

Speaker: Right. Right.

Speaker: I mean, think of all the movies that you see and it's really, you know, if you have eyes in the sky, so to speak, that can see things and all of a sudden that communication is shared across a network. You were talking about on-the-fly updates. You know, all of a sudden these missiles or these guns or these radars are all of a sudden able to see more than they could've before. You know, there's an awful lot that can be done. It's really just read your science fiction.

Speaker: Yeah, we'll get--so it's pulling that, that unprecedented situational awareness. If you think of John Boyd's OODA loop, right, orient-

Speaker: Right.

Speaker: Or orient, observe, decide, act, that's the decision cycle. We always say we want to be inside the adversary's decision cycle. So the sooner we can gain good situational awareness and observe our environment and then if we have some decision support automation, right, to make good, a recommender system. You know, shoot that guy, not that guy. No.

Speaker: You shot him.

Speaker: Yeah.

Speaker: So a great comment--

Speaker: Yeah.

Speaker: --From the audience, saying, "War fighters complain about situation clutter."

Speaker: Right.

Speaker: What's the comment back to that?

Speaker: Oh, that we can use--we could use automation and machine learning in particular and decision support systems to reduce that cognitive load and remove a lot of that data clutter, and actually let it organize and categorize and classify it behind the scenes.

Speaker: Right.

Speaker: Okay.

Speaker: I mean, that's what machine learning does great is it classifies really well.

Speaker: Right. I'd say the best strength there, the bare bones machine learning, technology that's been available for almost 50 years at this point, if not more, humans can look at some data and find some patterns. Machines are incredible at looking through enormous amounts of data and finding the subtlest of patterns, and the benefit is the clutter that's being referred to in the

comment, I imagine, that type of clutter can be removed through intelligent A.I. and ML designs, artificial intelligence, machine learning designs, to give the--so the machine is looking at only the important stuff.

Speaker: And it never gets tired.

Speaker: Never gets tired.

Speaker: It's relentless. Yeah.

Speaker: Looks at more than you can look at ever.

Speaker: Right.

Speaker: It's--that's the kind of benefit, I would imagine.

Speaker: Right.

Speaker: Yeah, you don't have to feed it.

Speaker: I would even think--

Speaker: Doesn't get emotional.

Speaker: It doesn't get emotional.

Speaker: I would even think if there's physical clutter, maybe, you know, something like AR or something, so you're actually looking at your surroundings, but it's only highlighting the important things to you.

Speaker: Right.

Speaker: And something like that I think would also be enormously useful and--

Speaker: Yeah. And we had that one research project from last year that they worked on about--

Speaker: Right.

Speaker: --Augmented reality.

Speaker: I think they call it CAVIAR.

Speaker: Caviar, yeah. Cyber Affordance Visualization in Realistic Environment, something like that.

Speaker: Right.

Speaker: But it was really, if I look around my environment, I want to be able to highlight things that are important to me in my mission.

Speaker: Right.

Speaker: So if I'm interested in weapons of mass destruction, it would highlight, you know, trace elements of bad chemicals or something. If I'm interested in, I want to do a network infiltration, it would highlight wireless access points or Bluetooth online--

Speaker: Right.

Speaker: We've actually already seen that with the project you had a while ago on looking at information coming out of just skin. You can see heartbeat and heart rate.

Speaker: Right. Oh, absolutely, yeah. So you can just look at heartrate from, you know, actual camera, video, that's being captured, and so you can look if somebody's getting tired and/or if somebody's getting, you know, I think PTSD is a big thing, right?

Speaker: Mm-hm.

Speaker: So you could look at all those things and so I think machines can really help even the human element in the battlefield. Especially if they're facing all kinds of, you know, weather elements like, I don't know, cold or heat or something like that. You can actually see on their skin if they're being adversely affected by things like that, like, you know, hypothermia and stuff like that, so...

Speaker: Yeah. And the traditional answer of the three "D's", right, dull, dirty and dangerous. Take people out of dull, dirty and dangerous positions.

Speaker: Right. Right.

Speaker: Right.

Speaker: So other questions from the audience?

Speaker: Just another comment. It may have been coming in when you were talking about the whole artificial intelligence thing. Said, "What percentage of war fighters are killed while looking at their smartphones?" You know, it's a--

Speaker: I don't know the answer to that.

Speaker: Yeah.

Speaker: So it's interesting, because I would phrase it differently. I don't know if they're looking at their smartphones.

Speaker: Yeah. Yeah.

Speaker: But there is definitely a problem that there's a lot of information that comes in too, ranging from the top general down to the guy who's on the field, that he has to manage, and as machines get more and more intelligent, more and more capable, there was an old joke that some guy was trying to fire his AK-47 or whatever it is and they got a blue screen of death from the old Windows XP and it's like, "Reboot your gun." So there's a lot of information to be managed. A lot of, I think, what we're discussing here is trying to bring the capability of this automated filtering. Imagine if you could have, in a military sense, the same kind of sort by important that you have in your mail right now.

Speaker: Right.

Speaker: Yeah. So things like decision paralysis are a real thing when you're given too much information, and you can't decide what's important and what's not and you can't sift through it all, and so the ability to use automated systems, whether it's machine learning or not,

to categorize and separate that information and help combine it and make sense of it, I mean, we've got to go there. Or we're going to, let's just go back to Game of Thrones and we'll all use swords and--

Speaker: Hacking swords.

Speaker: --And dragons and--yeah.

Speaker: I think we got to listen to the war fighter too. Find out what the problems are and--

Speaker: Yeah, absolutely.

Speaker: --Reiterate on what we know.

Speaker: Right.

Speaker: Yeah. And so, yeah. To bring it back into the DevOps principle really too, is integrating the actual users of the attack right into the development and the experimentation and the prototyping and knowing that we're not going to get it right the first time. We have to continue to experiment prototype, etc.

Speaker: So we got about a minute left, so anything just to wrap up the conversation today, Jeff? I know we're talking about maybe producing some other content throughout the year to talk about these topics.

Speaker: Yeah, yeah. I have a weird one and then I'll let everybody else to have a final one.

Speaker: Yeah. Yep.

Speaker: Talking about resilient design and fail-safe design, I think one of the most elegant designs ever is the escalator, because when it breaks it's a perfectly functional set of stairs.

Speaker: Right. Yes.

Speaker: So it's a great design, yeah. So let's go this way. Joe, anything else?

Speaker: No, that's it. Stay classy, Pittsburgh.

Speaker: Yeah, how do we hear? Go Steelers.

Speaker: Yeah.

Speaker: Looking forward to keeping--do all the good work we do to bring this stuff to the DoD.

Speaker: Okay, Grace?

Speaker: Same here.

Speaker: We're getting the "time's up" signal.

Speaker: Same. Same.

Speaker: I want to thank everybody. I want to thank the audience for listening. I want to thank the folks on our staff, Shane and everybody, for putting this together. There's a lot of work that goes on behind the scenes to pull this off, and thank the panel members for doing this.

Speaker: Yeah. Thanks for everyone time. Then we want--

Speaker: And thanks for, Joe, right, for the shirt?

Speaker: Yes.

Speaker: Yeah.

Speaker: Thanks, Joe, for the shirt.

Speaker: And thanks everyone for attending today. We will look to get a cleaned-up version in the archive for any audio that was missed at the beginning. But we appreciate your time today, and that's going to wrap it up from here in Pittsburgh, P-A. Thanks, everyone. Have a great day.