Is Software Spoiling Us? Innovations in Daily Life from Software

featuring Jeff Boleng, Grace Lewis, Eliezer Kanal, Satya Venneti, & Joseph D. Yankel

---

Bill Thomas: 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 and operated by Carnegie Mellon University. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

In this series of podcasts we are presenting excerpts from a recent SEI Virtual event, *Is Software Spoiling Us?* Jeff Boleng, acting chief technical officer, moderated the discussion, which featured a panel of SEI researchers: Grace Lewis, Eliezer Kanal, Joseph Yankel, and Satya Venneti.

In this segment, our panel discusses awesome innovations in daily life that are made possible because of software.

Jeff Belong: I am going to start with Dr. Grace Lewis. She is 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 a question. We are 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 government life, but in your daily life, some awesome capability achieved through software?

Grace Lewis: To me it is 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.

You can imagine a scenario in which a user has a smartphone, and the user says, *OK, time to go shopping*, right? 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. 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 am missing.

Jeff: Do you have a smart pantry?

Grace: I am working on it. 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. 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*. 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 are cheaper, it can buy those and pay for it with your credit card. 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. They are all within a 10-mile radius of my house because that is what I told it to do. This is not science fiction. It is something that is perfectly feasible today.

What are the technologies that are enabling that? Well, there are lots of them. 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 is out there. So not only thinking beyond just looking 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 all of a sudden you have a smart pantry.

Going back to your question. Virtualization, microservices. That is 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. 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. It's just amazing to me.

Jeff: Yes. You truly get penetration with a technology when it becomes ubiquitous, and you forget that you are interacting with it. Thanks, Grace. That was the highlight of dozens of cool things we are leveraging.

SEI Podcast Series

We are going to move to Satya Venneti. She is a senior research scientist here at SEI. One of her primary areas of focus is machine emotional intelligence or having machines help understand the state of the human being. So Satya, give us some awesomeness that software brings to us.

Satya Venneti: I'm really excited about machine emotional intelligence. Really I think of it as a shift from the information age to an actual communication age. Machines are able to detect, understand and respond to human users' emotions in real-time. That can actually help in many ways. I listed four here, which is Human-Human teaming. 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. They have these ID badges, which actually have sensors on them. They are in real-time able to transmit stress patterns and movement and so on. They actually help to understand how employees are engaged and improve processes and so on. They actually are using it on Wall Street to actually look at how their bankers are being stressed out. So I think that's amazing.

The other thing is Human Machine Teaming. Again, there is this startup called BRAIQ, or B-R-A-I- Q. They have these self-driving cars, and they equip them with sensors. They are actually looking at how people feel when they are in these self-driving cars and actually looking at how comfortable they are when they are accelerating or braking or so on.

The other one is augmenting human capabilities. So again, there is this startup from MIT, and it looks at speaking patterns. It is actually able to help call center agents to engage better with their customers. Like, *Hey, you are speaking too fast*, or *You are actually speaking over each other or You are interrupting each other*.

And then the last, and my favorite, is offsetting human limitations. There is 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. Teen-girl suicide is, I think, at a 40-year high. This is something that, I think, 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.

Jeff: What kind of technologies power all these innovations?

Satya: There is something called passive biometrics. It is this new generation of biometrics where we actually are able to look at people. Without contact, we are actually able to see and collect biometric information in real-time and actually analyze that in real-time. The drivers are that on the front end we have these low-cost, high-speed, high-resolution sensors that are available. On the back end we have new technique software, 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. I think this is the main driver for this whole machine emotional intelligence.

Jeff: Awesome. Thanks. We are going to go to Eli Kanal. He is a tech manager in our CERT Cybersecurity Division. 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 are going to get through all four

of us on this question. What are some examples of awesome things software has brought to society?

Eli Kanal: So there's a couple. One of my favorite ones that I will 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.

We are starting to see 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.

Jeff: I read that the win for AlphaGo was a strategy that no human had seen before.

Eli: 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. Any of the people who are watching it, they are wondering if it was a bug. 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. It is a pretty impressive achievement.

Jeff: That is topic's enough for a whole other webcast, but it brings up all the questions like, *Is it repeatable now*?

Eli: It is interesting. Let's get to that. One of the other areas where I think we have seen some extreme impressive advances is in the areas of self-driving cars. People do not tend to think of it, but a self-driving car is much more difficult than just, *Put a train on the track and, put a robot on the speed pedal*. There is an enormous amount of freedom. The car could drive on the curb.

There is an enormous amount of social mores to have to take into account. If a person is coming down the street, a human driver will slow down a little bit to indicate they have seen the person, even if they don't need to slow down. That kind of behavior is very difficult to train into a machine. The ability to do that has really it shows kind of how far software has come that we are 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 is an absolute ton of them in the field of machine learning. To focus on one for a second, there is something called a recommender engine which can look at other types of, *what behaviors have you done before?* Based on what you did before, we were talking earlier about shopping. *Based on what you've done before, it looks like you might like this thing.*

You've never bought that before. 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. These advances and the way they've implemented them have really kind of brought some of this stuff into the next level.

Jeff: I love recommender engines because I hate shopping. 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. I love recommender engines, actually.

Last time for this question. We're going to go to Joe Yankel. He is a senior software engineer in also in our cybersecurity division, but he really focuses on Secure DevOps. I think Joe is going to tell us about how DevOps saved healthcare. No, I'm kidding.

Joe Yankel: Well, actually, I recently did read 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 is now the head of a new department established in the White House, digital services. He was brought in. Healthcare is a big, big thing, right? Health reform hasn't been really attempted in decades. The 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, 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.* There his thought was, *Well, let's just 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 are eight million people enrolled, big success.

We do focus on DevOps. It's unique for every company. There is 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. It involves quite a bit of communication.

Communication is key. It is something that we have 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…

Eli: Yes. 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, and new requirements. Our focus has been on Agile and DevOps, and it does allow this to happen.

Jeff: Yes. I'm going to not let myself off the hook. I've got to talk about something that I think is awesome that software achieved. So I have two examples, actually.

SEI Podcast Series

A lot of people that know me have heard me tell this; 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've ever been in a car with adaptive cruise control. It is 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, silently for the night, an over-the-air software update happened that gave them like almost complete self-driving capability, at least on the highway, anyway. It is 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 am just floored by that, that we are able to do that now. No technician came to the house, no new sensors were added, no new hardware, 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. If we all remember the Flappy Birds application. He wrote that thing in three days

So 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 did not cure cancer, but it had significant global impact. There was probably an awful lot of productivity lost because of Flappy Birds at work, right. I'm going to save my answer for 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 at easier.

Bill: Thank you for joining us. Links to resources mentioned in this podcast are available in our

The complete webinar, *Is Software Spoiling Us?*, is available in its entirety on the SEI's YouTube Channel.

This podcast is available on the SEI website at sei.cmu.edu/podcasts and on Carnegie Mellon University's iTunes U site and the SEI's YouTube Channel. As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu. Thank you.