DOD Software Modernization: SEI Impact and Innovation

Featuring Paul Nielsen as Interviewed by Matthew Butkovic

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

Matt Butkovic: Welcome to the SEI Podcast Series. My name is Matthew Butkovic. I am the technical director of Cyber Risk and Resilience in the CERT Division of the Software Engineering Institute. Today, I am joined by Paul Nielsen. Welcome, Paul.

Paul Nielsen: Welcome. Nice to see you Matt.

Matt: Good to see you. Paul is the director and CEO of the Software Engineering Institute. He served for 32 years in the U.S. Air Force, retiring as a major general and as the commander of the Air Force Research Lab. Today we are here to talk about the SEI support and our vision for sustaining software modernization.

Paul: Software modernization is so important. This is something that has been a highlight during my career, including my Air Force time. I entered the Air Force in 1972, so that is a long time ago. At that time, software was sort of a minor part of many systems, but during my career, it became a bigger and

bigger part of most of the military systems. Over the last 20 years, it has become such a big part of all life in the world, basically.

Matt: Software is the heart of everything. I think you can agree. You have had the privilege of seeing software evolve in many phases. One of the things that we can be assured of is that software will continue to evolve and the need to modernize and contemporize what will be one of those facets of our experience is going to be enduring.

Paul: That is true. One of the big things there is that as the systems have gotten larger, more complex, and more interconnected, it has become even more important to modernize them throughout time. This is an area where the government at times has kind of lagged the commercial world because, most recently, some of the large software companies have developed techniques to keep their software on the edge all the time.

Matt: Paul, the need for that cross pollination and collaboration, it seems to me that the things that private industry does in software and the things the government and the Department of Defense do are converging.

Paul: We hope so. We hope so. This is where spin in from the commercial world could really help us.

Matt: Well, that was really the question I was building towards, which is how do you think we can assist in in generating more substantive collaboration and bringing in the best practices from industry into the DoD and the federal government software process?

Paul: This is really a special mission for the Software Engineering Institute is to be aware of all the different techniques being developed in the world and to bring the best into the Department of Defense. Now, the Department of Defense has some special features that make it a little different than the commercial world. We actually build systems that are designed to be lethal where most games are not meant to be that way. But the techniques that have been developed by the commercial world over the last 10-to-15 years of iterative development of increased security in certain areas have really made a difference and have made it so software, as the Defense Innovation Board said, software is never done. It's always evolving, always becoming better. That is the best thing to have happened.

Matt: The idea that quality is an emergent property, right?
It's never finished. It's really interesting. So there was a report from the

Government Accountability Office in 2023 that said that of the $100 billion that is invested in IT and cyber-related activities, 80 percent is earmarked for maintaining legacy systems. Thoughts about how here at SEI we are coming up with innovative solutions to reduce and maybe change that ratio?

Paul: One of the things that is very difficult for the government is to really terminate systems. Usually they build a new system, but then they keep the old system going. During my military career, I worked in Cheyenne Mountain. When I was in Cheyenne Mountain, we had the new system, we had the old system, and we had the original system. Everything was considered a backup. It is expensive to maintain these systems with time. Really you have to get onto the path where at some point you sunset systems. The commercial world has learned how to do that quite well.

Matt: That agility seems to be one of the one of the central hallmarks today in modern software development and operation. You have referenced DevOps and applying those sorts of methods. I know here at the SEI we have had a long history of being in the forefront of that. From your experience in uniform, are there things that are unique that need to be considered? Do we have to tailor those approaches for the military context?

Paul: I already mentioned the fact that some military systems have lethality in them. You want to be extra careful. You probably are going to do a little extra testing on systems that that are designed to be lethal. But when you look at the vast array of software systems that the government uses, they have all different kinds. In the Defense Innovation Board study, they talked about four types of software, commercial software running on commercial systems that maybe aren't unchanged. The Defense Department has many of those things. A lot of the accounting systems, a lot of the personnel systems, could be the same as the commercial systems that are out there today. Then you have software that maybe is customized a little bit for defense, but you still want to hang tight to the commercial version as tight as you can. Then you get to things that are a little more customized that are running on maybe even special hardware. When you get to the point where you are doing customized software on customized, unique hardware, that is a little bit special sometimes. The flight software system is running inside an F-22 or F-35 or the systems that help aim an Aegis cruiser's guns, these are different systems than the commercial world has. Sometimes there are special things that have to be done for them.

Matt: It is a really important point that the software that we are referencing in our discussion today really does span the gamut from embedded weapon

systems to things used for payroll and record keeping. And different attributes matter. In those contexts you mentioned lethality, so there is ensuring that you apply lethality, and there is also the safety-critical nature to make sure that we are doing this to a controlled and safe manner as well. And all of this is underpinned by software. I think it's fair to say that software is at the heart of everything, as we have said, but also the nexus. Here at the SEI we have the teams focused on AI and on cyber. I was wondering if maybe you could just speak to the join between those things if we think of a Venn diagram of those three cases.

Paul: These three things do interrelate so strongly, especially as we go forward. I already mentioned the fact that software itself has become larger, more complex and more interconnected. Because it is interconnected, any vulnerability anywhere in the system can be a vulnerability to everybody, right. So the software cyber connection has to be very strong. But in addition, now that we are bringing AI into the system, AI can help with software development, it can help with security, but AI introduces its own vulnerabilities. It is a bigger emphasis on data. There are ways to poison the data or have incorrect, have biased data maybe compared to what you suspect that can make the system not work the way you intend.

Matt: The introduction of AI automation, machine learning broadly, means that we can do things at a scale and a speed and perhaps at a reduced cost that we could have never envisioned 20 years ago. The flip side of that is the attack surface is expanded.

Paul: Yes, the attack surface keeps getting larger.

Matt: Right, so now we have a host of new worries, and striking that balance I think is one of the key obligations we have here when we consider that the joint between cyber and AI and software.

Paul: One of the big challenges of engineering ever since engineering was invented is balancing because you have to have a reasonable cost, you have to have a reasonable schedule, but you want to do something new with engineering, you want to solve a problem that maybe wasn't solved before. One of the jobs of an engineer is to find that balance, and the balance is different, different times for different kinds of systems. As we point out, in some systems maybe safety is the key parameter you are trying to emphasize. Other times it might be the speed at which you can deploy something. So you really have to find a sweet spot in the balance equation.

Matt: As you said, software is never done. I think that calculus is never done. It is very contextual in my experience. Shifting gears slightly, Paul, you have had the opportunity to lead a number of seminal studies in software and software modernization. There are many references that we could make to the catalog of things that the SEI has contributed to. But when I think about the Defense Science Board study, I think that is worth spending a little time exploring maybe our participation and your leadership and the conclusions of that work if you are willing to unpack that for us.

Paul: Sure. Well, I was a member of the Defense Science Board for about eight years from 2013 to 2021. I am still helping them, but now I am a consultant on certain studies that come in because they turn over the membership. In 2015, I was the cochair of a major study that they did. We call them summer studies. They do one of those a year, and it was on autonomy. That was a really big thing to look at in 2015. Now we are almost 10 years past that, and it is probably time to redo that because so much has happened in the autonomous world. We see all the cars that are trying to drive autonomously. We see the drones that have certain degrees of autonomy. This is an important thing again, especially when lethality is involved. Now in the Defense Department, sometimes lethality is intentional. In a self-driving car, lethality is not intentional, but we don't want that feature to come in. The commercial world and the government world sometimes is starting to crossover. This is a place where the commercial world can learn some from the Defense Department on how to ensure the safety in some of these systems.

Later, in some of the other studies we have done, we did studies on software development. In 2018, the Defense Science Board did a study on software acquisition at almost the same time that the Defense Innovation Board, which was a new board, did their study on the 10 commandments and Software is Never Done. The two really came together well. It was a really special time in the department because I think the Department of Defense was becoming more and more aware of the fact that their own software practices were sort of lagging, and that maybe there was too much of an emphasis on regulation and requirement studies that in some ways, like the commercial world, it might be better to build some software fast and incrementally, get something out and then continue to refine it. That is where I think in the Defense Department, the acceptability of iterative techniques really went up. At the time, Will Roper was the Air Force acquisition exec, and he really pushed on that. This administration that is new is coming in is really pushing on that too, because they see what the commercial software companies have done. If you look at what Netflix, Amazon, Google, Tesla,

SpaceX, what they have done, they have largely accomplished what they have through iterative techniques. Build something. Build something fast. Test it all the time. Build an environment that is very supportive and then continue to improve it on a weekly, daily, sometimes hourly cadence.

Matt: So really changing our mindset from projects and development processes that are monolithic for—it sounds pejorative and maybe it should be—but things that are vastly more complicated and time consuming than maybe these smaller iterative approaches.

Paul: Early in my career we started to build these bigger systems, and one idea was that you had to, how to really get the requirements straight. I think we carried that too far where we got to where we might study requirements for five years or six years. During that time, the technology base changed several times, and the threat changed several times. And so how good could that be? It is much more important to get some working software out there soon, but have it architected in a way that it can continue to improve.

Matt: One of the things I am really proud of working here at the SEI is the work that we do is enduring. As the technology stack changes or the code base changes, the fundamental principles that we have developed and transitioned are that through line. I think that is a really good example where we are going to see new programming languages, the introduction of AI, we are going to see deprecated software removed, the stack is going to change, but if we focus on the fundamentals and measures that we know work, I think it is a great way to not only jump start that process and maybe ensure that we are rationalizing requirements but have verifiable confidence in the systems that are being operated.

Paul: Another thing I am very proud of is this was actually a National Academy study we did looking at the software professionals inside the government, because the government does have software engineers, especially in the sustainment world. In that study we made some recommendations that have been largely adopted so that inside the sustainment world—Army, Navy, Air Force, Marine depots and such—they are largely moved over to a continuous integration, continuous development and deployment kind of system. This is very important because sometimes they are actually maintaining code that we would call orphan code. This is code that maybe was originally written 30-40 years ago, and nobody else is maintaining that now except inside the government. Now, we would like that code to be modernized, but sometimes there are reasons why you are not modernizing it yet. It might be that the hardware it is written on is also very

old. These people have taken the newer techniques and applied them to these old orphan systems, and they are making great progress on that. So I am very proud of that.

The other thing that we have done to really help the Defense Department is the whole idea of these software factories. One of the items in the 10 commandments that the DIB came up with was that you do have to have an organic force. Even though a lot of work is done by contractors, you have to have an organic force that is close to the user that can continue to modify and iterate software so that it stays current for the soldiers, sailors, airmen, marines, guardians to be able to do their job. You want to be able to evolve the software in a faster OODA loop than your enemy can in a sense too. So you want to be able to continue to bring new capabilities, not on big block time frames, like every 20 years, but on weekly, monthly time frames.

Matt: That I think is one of the most important distinctions we can draw is that the time to react in all contexts has just been modified. This will continue to accelerate as we move to more automation and the application of AI, we will reach a point where human-in-the-loop might become one of the barriers. But I am glad you mentioned the OODA loop. I am sure John Boyd never thought that that would be discussed in this context, but I am glad it has come up. You also mentioned the workforce, and that is really important. The Defense Industrial Base supporting the EEOD needs to ensure that their folks are perpetually keeping pace and hopefully leading the charge with new techniques. We have had a very important hand in that over the years, as you are pointing out, both in software, cyber, and now increasingly in AI. So thoughts, Paul, about workforce development?

Paul: That has always been a mission for the Software Engineering Institute. We developed some of the curricula used in colleges, junior colleges, associate colleges and stuff for software engineering cyber and kind of working on that in the AI world right now too. This is very important because you have to have a workforce that stays up with the times.

Now, when you look at the workforce that is involved in defense, there are several different communities. You have the uniform group. They turn over pretty fast in their jobs. You have the civil servants who tend to be in positions for long times and have some stability. You have the Defense Industrial Base, which maybe has contracted so much that maybe it is not as competitive as it once was. Excitingly, you also have the commercial world. One thing that has been a prime initiative for several years now is, how do we get the non-traditional defense contractors involved in defense work?

That includes the big companies, the ones that we hear of all the time, the Facebooks and Googles and Microsofts. But it also includes all the very small innovative companies. In the AI world, you see a lot of these small companies rising very fast. I mean, OpenAI, who knew that name even five years ago. Anthropic, who knew them? One of the great things about our economy, our U.S. economy, they help bring innovation to the mix. We want to encourage that innovation to the mix. That is one of our special sauces.

Matt: That is of prime importance in all of this. We are located today in Pittsburgh. Pittsburgh was part of the arsenal of democracy, right? The industrial might of this city in this country was used during the Second World War. I know that that was predicated on the ability to bring in non-traditional participants into that ecosystem. I think it is very important as we talk about software that the balance between regulation requirements and innovation is appropriately calibrated, so that there aren't undue burdens to new entrants.

Paul: We don't want to have inhibitions to these new things. We want to facilitate the new ideas coming in. There is the book that came out several years ago, *Freedom's Forge*, which talked about the efforts in World War II and how we turned this economy around and traditional companies all of a sudden became defense contractors. Ford and GM and all these companies did. We want that to happen in software as well because we do have great software talent in the U.S. When I say software, I am speaking in a generic sense, software, cyber, AI. The advances in AI offer some amazing possibilities inside of our developmental processes and everything. AI systems can be the peer programmer and can help make sure that even as a person is typing in the code, someone is looking over the shoulder. AI systems can help with iterative developments by helping to run the test every night, every day, on every version of code so that we can make sure we have good regression tests and can move forward. AI can also dream up new possibilities. Sometimes people disparage about the hallucinations they hear about some AI systems, but sometimes those hallucinations might have the seed of a new idea. Everyone talks about the chess capabilities and then the Go capabilities. I think it is in the Go arena that an AI system did some moves nobody had ever seen before. All of a sudden they go, *Wow, this is not just emulating what great masters do, it's innovating what great masters should do.*

Matt: Fascinating, right? This is where we are seeing the technology outpace our ability to even comprehend its possibilities. I want to shift then slightly to ways we have tried to capture what we think is coming next. In particular, Paul, in the domain of software and software modernization, the National

Agenda for Software Engineering report that was published by the SEI has a path and a suggested set of evolutions that will occur. Thoughts about that in this context?

Paul: Well, I was really proud of that. We did that kind of as a service to the community. It is not just our agenda, it is a national agenda, perhaps even a global agenda for the work that needs to be done in software as we go to the future. I think it points out that there is sort of traditional software engineering that can do so many things for the AI world, but also that the AI world is starting to change software engineering and starting to bring in new ways of reuse of software and new ways of modifying software as we go forward. That is really exciting. Since that report came out, I know when I have gone to a few of the big worldwide conferences, ICSE [International Conference on Software Engineering], it has been referenced time and time again. This was a great effort by SEI to serve the whole software engineering community on where things should go.

Matt: And a great example of the thought leadership demonstrated by the SEI drawing on the community for the community and leaving an important artifact behind. If we think about future innovation. Paul, a question about where…It is fraught with peril to make predictions.

Paul: Yes, it is.

Matt: And I have one final prediction at the end of this for you. But if you look 10 years in the future, what topics do you think folks will be talking about? What would be front of mind in software, cyber, and AI?

Paul: Well, certainly the applications of AI are going to continue to grow, and the capabilities of AI systems are going to continue to grow. Right now, we are sort of in the teen years, not the baby years, but the teen years.

Matt: The difficult teen years?

Paul: The difficult teen years, yes, but there is more to come. One of the things that has happened in the LLMs, the LLMs know past knowledge well. Sometimes they get dated, and we have to find ways to have the LLMs kind of continuously pull the latest information to make sure they are doing the right things as they make some recommendations.

I do want to get back a little bit to the idea of the innovation side of AI. These LLMs…I know an organization that was using them at one point to suggest

some titles for an article. None of the titles that it suggested were the ones they ultimately used, but the editor said there were things in their suggestions that led to the final answer that they did because they hadn't thought of it that way. Sometimes the idea that they are hallucinating make it, like I said earlier, a germ of something really good in there in the future.

Now as we look at the future, quantum is coming down pike. It is still not there. But quantum computing, quantum cryptography, quantum everything is coming down pretty fast right now. I think one of the things that the DIB mentioned was the fact that we need to have computational power available to everybody to be able to do some of these things. That is going to be true for quantum too. As we go forward, the more that we can have computational power available to the masses, to everybody, the more innovation that will occur. So I think we need to kind of continue to develop ways to have more computational power available for everyone.

Matt: That is a really interesting point, that we know quantum is going to reshape everything, literally completely reset our expectations about computer architecture. At the same time, we will have an even greater demand for ubiquitous computing. No matter what shape this takes, I am confident that our organization will deliver innovation and the thought leadership required to navigate those new challenges.

Paul, I have a question for you. We have a contest coming up on Sunday. OK, And the final prediction, future prediction. We have the Kansas City Chiefs playing the Philadelphia Eagles.

Paul: Yes.

Matt: Thoughts.

Paul: Wow, two great teams, two great quarterbacks. A little bit different approach because the Eagles have this great running back and Saquon Barkley, a Penn State guy. They each have been innovative in the way they have brought some things to the table. The running game of the Eagles has shown some real innovation. Mahomes is such an innovative passer and signal caller. It is going to be a tough game. My heart goes out. I love the Chiefs, but my heart goes out to the Eagles because I like to see a new person get in the game here. I am pulling for Jalen Hurts right now.

Matt: I am as well. Thank you for offering that analysis. To tie it back to the topic at hand, to make this seem like it wasn't completely extraneous,

innovation, right, whether it is the brotherly shove or running new offensive schemes, innovation in all facets of sports and industry, and our role is really defining hallmark of a successful organization.

Paul: Yes. Innovation can come from all different quarters. Sometimes it is easier for a young rising scientist to innovate because they haven't run into any obstacles before, and so they will dream big. Think of Einstein in 1905 and all that stuff, but sometimes it is the older, more mature people, who innovate as well.

Matt: Exactly.

Paul: So don't count us out either. You really want some ideas coming from all quarters if you possibly can.

Matt: I wonder when the NFL will specifically prohibit using AI to develop plays. That is probably not that far away.

Paul: They will probably let them innovate on plays because how can they stop it right now, but they probably won't allow them to play.

Matt: Yes, Paul. Thank you so much for joining for this conversation.

Paul: I really enjoyed it.

Matt: For the audience, we'll include links in the transcript to the resources mentioned. Join this podcast and on a forthcoming blog on this topic. Finally, a reminder to our audience that our podcasts are available pretty much anywhere podcasts are available, SoundCloud, Spotify, Apple podcast as well as the SEI's own YouTube channel. If you like what you are seeing here today, please give us a thumbs up. Thank you for joining us.

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