# EPISODE 1115

[INTRODUCTION]

**[00:00:00] JM:** The U.S. Army Cyber School is a training program which trains cyber soldiers and leaders to be adept in cyber military strategy and tactics. In order to teach these skills, the Cyber School uses a system they call Courseware as Code, a workflow that allows updates to the curriculum in a reversion-friendly fashion similar to infrastructure as code.

Ben Allison teaches at the U.S. Army Cyber School and he has put work into developing the training program and the ongoing lesson plans. Ben joins the show today to talk about how the U.S. Army manages curriculum through Courseware as Code, as well as the work that he has done to improve this workflow overtime.

Ben is also speaking at GitLab Commit 2020, which is GitLab's upcoming conference. You can register for GitLab Commit yourself by going to softwareengineeringdaily.com/gitlabcommit. Thank you to GitLab for being a sponsor of Software Engineering Daily. And I hope you enjoy this episode.

[SPONSOR MESSAGE]

**[00:01:00] JM:** When the New Yorker magazine asked Mark Zuckerberg how he gets his news. He said the one news source he definitively follows is Techmeme. For more than two years and nearly 700 episodes, the Techmeme Ride Home Podcast has been one of Silicon Valley's favorites. The Techmeme Ride Home Podcast is daily. It's only 15 to 20 minutes long and it's every day. By 5 PM Eastern, it has all the latest tech news, but it's more than just headlines. You could get a robot to read you the headlines.

The Techmeme Ride Home Podcast is all about the context around the latest news of the day. It's top stories, the top posts and tweets and conversations about those stories as well as behind-the-scenes analysis. Techmeme Ride Home is like TL DR as a service. The folks at

 

Techmeme are online all day reading everything so they can catch you up. Search your podcast player today for Ride Home and subscribe to the Techmeme Ride Home Podcast.

[INTERVIEW]
**[00:02:04] JM:** Ben, welcome to the show.

**[00:02:06] BA:** Great to be here. Thanks for having me here too.

**[00:02:08] JM:** We haven't done very many shows about military and the software used in the military. Could you explain how software is used in the military and give Git as an example of a piece of code that is used in the military.

**[00:02:21] BA:** In my particular experience, I am assigned as a faculty member at the Army Cyber School House. So my experience with using Git as a software development tool is primarily focused on that environment. And so the rest of the army, there are development organizations that use different tool suites, Git being on, GitLab, Atlassian, different development organizations across the DoD use different tool suites. But the Army Cyber School, we use GitLab and Git primarily as a way to manage our curriculum.

So the way that came to be prior to me arriving at the school around 2015, 2016 timeframe, the school first was created in 2015. So they were kind of – They got their first students – Created in 2014 for students in 2015. So they were kind of acting like a – They're very much a startup culture. So they had the flexibility because they had such a short timeframe to build their courses for their first students, which usually the army gives you three years. They had less than a year. So they had the flexibility more or less that the political flexibility to do what works and get things done rather than to wait on bureaucratic systems of what maybe in the past. So they had a lot of opportunity to innovate in a way that might not otherwise be possible in a more established institution within the army.

For them, they were looking at how can we manage our Courseware? The army traditionally has a three-year cycle that updates at a very slow pace. It's all using binary data formats, such as Word documents, PowerPoints. It get stored in a web interface where they upload the

documents and then download them when you need to instruct. For us, we wanted to be able to have more flexibility or we could manage Courseware and then applying Agile software development principles. And then also be able to update and manage the Courseware without having to go through the tedious process of using these other outdated systems that were designed for more static types of curriculum that don't change very often.

For us, Git was a natural choice for those of us coming into this school who also have a background experience in software engineering. So I am a computer. So I haven't worked in the army in the development role personally, but many have, and that influences the decision to use GitLab. For us, instead of using Word documents, we use markup languages to track our curriculum, and then we use the CI pipelines to build that curriculum. Same thing with infrastructure as code. We use different formats. For us it's key templates and OpenStack and then we use pipelines to deploy it. For us, that's the framework for how we chose to use Git primarily because it was available and we had the freedom to do so because the organization is just being stood up and the leadership was willing to assume risk by allowing us to innovate in ways that might not otherwise be possible.

**[00:05:10] JM:** Courseware is a term that this conversation is going to focus on. Explain what the term Courseware means and how it applies to this conversation.

**[00:05:19] BA:** In the army, when they have curriculum development, Courseware can be – I'll frame it this way. When you join the army, there are different specialties for different section of the army. So you have armor, would be people who drive tanks. You got infantry, the people who go and do the very typical army infantry things. You've got field artillery. They fire cannons. You've aviators who fly helicopters. They've got things like signal corps who does communication equipment. And the cyber range was created similar to the signal corps, where signal is more IT. Cyber is more focused on using IP-centric and RF-centric computing space to create offensive and defensive effects in support of the army and the U.S. Government, so the Department of Defense at large.

So what that really means for us at the school is we're developing courses to support soldiers coming into the army and the officers enlisted and warrant officers who need the technical

theory that supports all of the operational context. The operational context is all stuff that in the cyber range they can't really talk about what they're doing of how they're applying the theory in some cases. But for us, we are strictly worried about the technical theory that underlines all of the operational applications. For us, it's Windows, fundamentals, Linux fundamentals. Understanding operating systems and how they work. And then understanding networking, TCP/IP. The full stack of networking and understanding it enough to apply it. And then security concepts both from offensive and defensive perspective. Whether you're trying to defend an offensive actor or you're on the offensive side. You need to understand all of that theory that goes behind exploitation. How to defend? How attackers hide? And so on.

So for the school, when they talk about Courseware, they're talking about – For example, the cyber common technical corps is a module that every cohort, whether the officers who are enlisted are required to attend. So for them, this course is the overview of operating systems, networking and security. So the Courseware is the facilitator guides that go to the instructors. The student guides, they go to the students. And then the infrastructure as code that deploys interactive ranges for the students to work on from the classrooms, or in this case, during COVID-19, from wherever they're working from in a remote location over a VPN.

**[00:07:34] JM:** Okay. So you've mentioned Courseware. You've mentioned Git. And you're talking about curriculum management technologies like PowerPoint, and Microsoft, and PDF. These seem like separate worlds. If you're talking about office management tools, that's at a higher level than something that you would need to do version control on. What's the relationship between the version control stuff and the traditional Office Suite?

**[00:08:04] BA:** Right. In the traditional army, this is something I would – For anyone who's ever been to a government school or [inaudible 00:08:10] government, you often – When you go to a course, it's kind of – The general stereotype is you kind of have a pulse and you sit in a class and you sit through PowerPoint slides. Then you enter some very candid questions that are checks on learning. And then you move to the next thing. And nobody every fails and it's not hard when you just exist and you get through it and then you go to your unit and you actually do your job and you're learning your job or your unit. That's the stereotype of how training is done

in the army. That's not always accurate of course. This is stereotype. That's how – In some cases it's like that's not how the army obviously [inaudible 00:08:42] and it's always improving. But that's a character of how it could be.

So when people talk about Office documents in the army, death by PowerPoint, sometimes there are certain models that are designed in such a way that the army says you're going to learn about this. And you sit in a PowerPoint and someone flips their slides and then you're done and you do for a certain length of time and they brief the slides to you, and you're done. The school, the cyber school, doesn't want to be slide-driven.
Instead of being slide-driven in PowerPoint, we instead are facilitation guides that are stored on markup. We can also do slides using Asciidoctor and Reveal.JS. They're technologies that you run through Asciidoctor, which takes the markup language and then spits out HTML in the backside. We do sometimes use slides, but we're not trying to be centered on PowerPoints. Instead, we're trying to use the software development principles to facilitate learning through hands-on interaction.

**[00:09:39] JM:** Give me a little bit more of a description for how a course within the military precedes.

**[00:09:46] BA:** Yeah. It kind of depends on where you're at and what the objectives are for the specific course. For some schools, you go in – We're on the Training in Doctrine Command, TRADOC. You'll come in, there's a part of the course that's kind of designed for someone coming to your specialty in the army that's designed. Everyone goes through the same part. So the common corps, it's the piece that everyone goes through. And then next you'll have a technical block that's specific to your branch. If you're on a branch that goes in the field, you'll have field training exercises that might take up the entire time.

In the cyber school, it's focused. We have the same common corps. And then they approach that with a more technical flavor to learn. And then they have a field exercise, but they're field exercise. And rather than they take on computers and they do things like mess with Wi-Fi and

RF these things out in the field. But then the technical blocks, for the cyber school, it's broken up into – We do, we do modularize it. So we take a module of programming, for example, and then we could give that to different cohorts and take that model and spread it around. So we're focused on the module within a specific course for a specific work role.

Same thing for the cyber common technical corps. That is a 9-week course where you would come and you report to the school. And as a part of your course, you go through different modules and the instructors will teach you. When it's not in a pandemic condition, you go into the classroom. Everyone comes in. You have classrooms that are connected to our virtual training environment. Inside of that network, it's a closed-off network. And then they can deploy the ranges. Or in COVID conditions, it's a little unique right now. So we have virtual conferencing through Microsoft teams or in our some other virtualizing – Or classroom virtualization technology, to make a virtual classroom in your browser and then you then have VPN access to the training environment again.

So you'll go through all the training. You'll go through a morning lecture of, "Hey, here is a topic we're going to talk about today. Today we're learning about operating systems. Today we're talking about windows. Here is a discussion of the boot setup of how Windows launches from all the way booting up. Here's how the different between the kernel and the user space." And then they'll have a graded activity in the afternoon. That's facilitated through a gamified environment. For us, in particular, in this course, I'm talking about CCTC. They use a capture the flag webinar-phase to give exercises inside of the virtual training area that correspond to whatever the lecture was for that day. So they get the theory and then they get the hands-on application. And then they do that iteratively throughout the entire 9-week module. So it's the same thing for that module. And then there's a programming module that's two weeks and there are other different courses that get pieced together in that same fashion in the technical blocks.

Then when we use – We're talking about Courseware as code. That is the part where it enables us to deploy that in a backend without necessarily having to manually configure all of those ranges. We just write it once using the infrastructure as code and deploy it to the backend.

**[00:12:48] JM:** Again, there's infrastructure as code. And you have this term, Courseware as code. Can you talk a little bit more. What do you mean by Courseware as code?

**[00:13:01] BA:** In this case, I guess contrast it again with, like you mentioned, PowerPoints and those other things. Rather than tracking that. If you made a PowerPoint and you had 10 slides and you wanted to change one of those slides and you want to see who changed it and when they changed it. We wouldn't be able to do that necessarily. You could potentially do it on a share drive or some other version controlled system. But PowerPoint isn't really designed for version control.

If you're saving in Git or a version control system that's based on text, then you have to – You'll get a binary blob, because PowerPoint is essentially a zip file underneath and you can't see changes at any granular level between files. It's going to take a lot of disk space.

When we saw Courseware as code, you still have that Courseware. The Courseware would be any instructional material necessary to teach a lesson objective. And lessons are broken up into – They call them [inaudible 00:13:47] learning objectives and learning steps. So kind of your outcome and then the steps to get to that outcome. Those all come from – All the material necessary to teach that outcome is the Courseware. So for the cyber school, we want there to be interactive classes. We have both the theory. Instead of using PowerPoints or Word documents or PDFs that the instructors would refresh on. Here's the guide for the instructor. That would then be tracked in Asciidoctor, which is a markup language or a markdown, which is in the markup language where the Courseware is the instructions for the instructor or the guide for the students so that they can reference. It's the website that would provide all of those guides to them, whether if it's in a web format. It's the slide decks that might be shown to the students if there are slides. So all the Coursewares, all of those components to support the learning objective.

So we track all of those in GitLab. And so when we're tracking in GitLab, we don't want to use traditional formats like Office and PowerPoint and all of those type of formats. We want to be

able to do it in a lightweight and we want to able to have it in a lightweight text editor and then see by line version control of one thing has changed.

So the code in this case is the code products are predominantly Asciidoctor, markdown, pipelines to convert that into other formats [inaudible 00:15:04] websites. And then additionally, there are – Like I said, the Coursewares. That would be then a cloud in each script. So basically when you build a virtual – You define a virtual machine template in YAML, which is a template. It's another language to define. It's often used for infrastructure as code. You define your network and then you would later then define the user data fed into each virtual machine. You could say you have a network with three subnets and 10 virtual machines and I'm going to feed the user data into each one. So the Courseware would be, in this case, an exercise that the students go to those virtual machine and complete task to find a flag in the virtual machine. Or if you connect to it through several hops, if you're learning how to tunnel through networks or whatever the exercise might be. The Courseware in that lesson is – The instruction material for the instructors. The instructor guide, this guide for the students. And then prompts and exercises for the students. And then finally, the infrastructure as code necessary to deploy that exercise. So an instructor would build it once and then it's deployed and we can deploy it scale for 20 students. So then we never have to worry about the person who designed it if they move in the organization or they quit. We don't have to worry about, "Well, they knew how to design it and we can't figure out how to design it again." We don't have to worry about students having unequal environments. Everyone has the same environment because it's all deployed from the source code. And if we have to make a change, we'll change the line in the source code and can see who changed it and why they changed it. We get centralized control. Or that approval process because you have the main branch and then you'll have your development branches and they make changes and they approve it by some approval authority. So that all packs together as the Courseware.

[SPONSOR MESSAGE]

**[00:16:54] JM:** As your infrastructure grows, you have SSH hosts and Kubernetes clusters in staging, production, and maybe even on edge locations or smart devices! Implementing the best practices to access them all can be time consuming and security tools tend to get in the way of engineers.

I recently bumped into an interesting open source software called **Teleport.** It is open-source, written in Go, and is a drop-in replacement for OpenSSH. Plus, it has a native support for Kubernetes.

Built by our friends at Gravitational, **Teleport** provides identity-aware access using short-lived certificates with SSO, session recording, and other features that ensure compliance and audit requirements. But Teleport also prioritizes developer productivity and convenience because it's built by engineers, for engineers.

Go give it a try by going to try.gravitational.com/sed**,** where there are links to downloads, documentation, and, of course, the GitHub repository.

[INTERVIEW CONTINUED]

**[00:18:19] JM:** These are really interesting examples of training exercises. Can you talk to another training exercise?

**[00:18:25] BA:** Yeah. Time back to cyber common technical corps. This was seen as the most mature course in using Courseware as code. In the security module, which I had, I actually had the opportunity to write some of the days in that lecture. When I was a cadet at school, I spent a lot of time in doing capture the flags. Those type of competitions where you do intercollegiate capture the flag. Like NYU host one every year. When I did that, I worked on a web exploitation lesson.

I was trying to teach [inaudible 00:18:51], "Here's my understanding of the difference between a client side and a server side relationship." In my instance, I built the facilitation guide in Asciidoctor, and then I built a Docker container that would deploy into the virtual training area. And that would be – That's OpenStack. Essentially, a private cloud. We use templates to deploy that. Then the Docker containers give each student and exercise I've built where they'd have websites that they could do – They'd understand the client side vulnerability versus server side

vulnerabilities and things like that. So it's all timed back to open theory, applying that theory and exercise. And then they would – In this case, it was modeled after capture the flag. So once they exploited some – A cross side scripting or a SQL injection or whatever, however a scenario was designed. They would get a flag. So that was, for them, it's a one or two-day exercise. The intent there is not to create mastery, but just to help them understand the types of exploits and why it's important to apply defensive tools against websites for example.

**[00:19:50] JM:** Why is it necessary to have these kinds of training exercises? What are these training exercises preparing the military engineers to do?

**[00:20:03] BA:** Right. The cyber range has essentially two primary customers that we as the institute, the training institution support. At the joint level, there's the cyber mission force, and that supports the United States Cyber Command. And at that level, they are the ones who are in charge of providing offensive and defensive effect. So they're defending American interests and they're prepared to – in a state of war, they're prepared to conduct authentic operations. So they're kind of that in that staging, whatever that perspective where they're preparing for something in the future. Just like all the rest of the military, is more providing readiness. That's the same thing with the offensive force to provide a readiness capability.

So for them, our goal for that is they get to the operational units and they have very poorly defined problems. When I compare back to the traditional army, if you have an Abrams Tank that was created in the 80s. It's been around for about 40 years. Very little changes. If you're trying to train someone how to use that, it's very checklist-oriented. It's very established. Not much is going to change. When it does change, you know it's going to have well ahead of time.

Meanwhile, when you have the cyber forcer, you go out to the operation force. You don't really know what you're getting into, because one it's very new. Things change very often. And the problems people face are very dynamic. If you're asked to help defend and asset or a network or a piece of equipment or anything like that, you may not necessarily be able to train someone for every possible combination. But what you can do is train people how to solve problems.

So our primary goal with our curriculum is to give people interaction with solving problems where they're not necessarily – We're not necessarily giving them all the answers ahead of time, or just going to give them a problem, give them the opportunity to solve that problem. And then we're there as instructors to help nudge them along the way. So we're trying to build the problems solving that becomes muscle memory.

So then when they get to the technical force, the operational force, and they're trying to apply these technical concepts in a way that they may not have ever been taught, we can give them the skills necessary to find the answers themselves and become problem solvers. So that when they get to hard technical problems, they can solve them on their own even if we didn't necessarily have a foresight to know which technical problems they would need to solve.

**[00:22:22] JM:** So I can imagine that in this kind of world, you're constantly needing to adjust the curriculum. How does the rapid pace of curriculum adjustment map to what kinds of technologies you need to manage this Courseware as code?

**[00:22:42] BA:** Yeah. In this case, a lot of our changes that we get – I guess there's a couple you can answer this. The changes we make on the fly. So if we have a curriculum and we know something is wrong with it, we can make instantaneous changes. So if there's a typo and some sort of formatting, we can fix it. If there's a lane doesn't deploy properly, we can fix it. If we want to add depth of richness to a training environment, we can update the Courseware that way.

There's more official processes within the army to provide feedback from the operational force. We'll say, "Hey, you're not teaching these kills, and we think you should be teaching this." That's a three-year process. So the army – It's called a really terrible acronym. It's a CTSSB, or critical site task selection board or something of that nature. But they bring all these people in and they sit together for two weeks and say, "These are all the new things you need to learn." And then it goes back to the school house and they're supposed to update everything. And then within three years, have all implemented to do it again.

But that doesn't really work in a technical branch where three years is a huge length of time

when we're talking about information security. For the school, one of our intents with Courseware as code is to provide through GitLab the ability for our people in the operational force to say, they get to the unit, "Hey, I would love if you would teach this instead." So then they go on to GitLab in the public repos where students have access to and make issues and say, "Hey, we think that you should teach this." And so then while a student, a single student's feedback isn't necessarily a cause of change, anything, if we get enough feedback or if more than one person says it or if it's someone who's like a higher level representative. We could take that feedback, then follow along and approve a process where the course manager could eventually approve that issue and makes a change, or if it decides not.

And then at some point, if we make changes that are meaningful or change the outcomes, we would have to then mirror that with the official processes to approve it in what's called training and development capability, or the webinar phase I referenced before. That still tracks a lot of our curriculum for TRADOC standards to keep us approved under the TRADOC regulation. But we also then are able to – And our approach to do that is through tag releases. So when we get to a tag release, we'll push that tag release so it's back on the training development. And then we kind of iteratively improve little bits. Then once it gets to a threshold that the school set, we'll update the next one.

**[00:25:02] JM:** As far as an update, can you tell me about what kinds of Git workflows you're using? Who is involved in the review process?

**[00:25:11] BA:** Yeah, for us right now, it depends on the level of the change. For the way traditionally things work with software development. As most people – You're probably very familiar with it. You have your release branch and then you have development branch tied to issues. And we try to implement that as much as possible. One of the interesting facts that work of applying a software development principle when your instructors are not all software engineers is that doesn't always work in practice and people aren't always familiar with it. People aren't even familiar with sometimes with Asciidoctor. If they're good at understanding networks and teaching networking, they may not necessarily be a software engineer.

So when they come on board, we kind of get them up to speed. Explain that process and often times people fight us and they think it's dumb. And then by the time they've been here a couple years, sometimes people move on to other things. And then one of [inaudible 00:26:00] comes it's like, "Yeah, I got bit by Courseware as code. I love the way we do things here. I'm going to take it with me to my new job." If they stay in structure and things like that.

But the process for developing an approval process for this, to get back to your original question, is primarily if it's a low-threat issue, then it will go to one of our – We have instructors who are contracted. And then we have civilian instructors who work for the government. And then we have course managers. So the contractors will make changes perhaps and the civilians will approve it. If the government civilians and if it's a significant change and the course manager approve it, and if it's some other change, that's actually going to change the outcomes or the learning steps or things as we have captured. The definition of the training per TRADOC's regulations, the higher headquarters approves everything, then that has to go all the way up there. That's the part where once we get to that thresholds, it goes up to a – We can no longer use software development principles. It has to go through the normal army system. But we try to balance it so that happens, we limit the number of times we have to make changes that are so significant and it has to go up that. We try to do it right the first time and then minimize the number of times you make so significant changes. It's changing the outcomes or the resourcing or requirements or things like that.

**[00:27:14] JM:** On the surface, it doesn't seem that curriculum management would have a complex CICD workload. Can you tell me about how your pipelines, your CICD pipelines are taking center stage and the course content generation process?

**[00:27:35] BA:** Yeah, that's a great question. From the Courseware as code, from the content perspective, not the infrastructure. I kind of break it up in two. The Courseware would be the guides and all that. And then infrastructure is all the stuff that goes into our private cloud that we host in house. For the markup and all those language, Asciidoctor, we primarily use fairly simple pipelines that will take the curriculum and build it into static website. So we use two primary technologies for this. We use Antora, which is primarily based in Asciidoctor. So you build your

website and guide in Asciidoctor, and then it runs to a pipeline and deploys it to GitLab pages. That one is fairly simple. Not very complex. It's really just a build and deploy stage. Two stages in the pipeline.

However, there are more complex pipelines and such where the infrastructure as code comes in. So one of our instructors actually [inaudible 00:28:29] individual. He's a Marine instructor actually, because we have students who are in the Marines who work for us and we teach Marine as well at the cyber school, the Army Cyber School. He took it on his own. And when he saw the pipelines, he build a pipeline, what he calls the range deployment automation framework [inaudible 00:28:45]. But it goes through, I think, it's 7 or 8 stages and it pulls down the code and then it deploys the student ranges and then it takes the output of those ranges and it feeds into CTFD, which is a capture the flag framework. That's open source and it builds a custom interface for the students based on their deployments. The outputs of the deployments. So if it gets a DHCP IP address or gets a float IP and OpenStack, that gets fed into whatever gets deployed for the students. And then it builds exercise prompts for students based on what was deployed from the CI pipeline. And then it builds OpenStack, or builds CTFD in OpenStack. Then finally, it will go through – And some of this is still under development and how clean everything works perfectly, because we're still iterating on some of these more advanced uses. But it will then go through and ensure everything is deployed properly to make sure nothing fails. So all the activities are deployed. We want to go through it. We'll have a script go through and check each activity to make sure everything is deployed properly to make sure a student doesn't get to greater activity and it's broken and we don't know it. Since that deployment framework is the most mature implementation of using CI pipelines to deploy infrastructure.

**[00:29:59] JM:** Give me a little bit more of an overview of the tooling that you've used. You've now mentioned Git and GitLab and OpenStack. Tell me more about the tools that you're using.

**[00:30:10] BA:** OpenStack, it's a private cloud technology. When I mention OpenStack, with that comes all the supporting technologies, which is direct [inaudible 00:30:18]. You got your compute. You've got your networking and all of these different technologies under the hood, KVM, QEMU, which does your virtualization, your hypervisor. We also have a designate

included. We have containerization in our – I think we use Zun, which there's a couple of different containerization technologies and OpenStack. So we have all those to support the private cloud piece.

Within our classrooms, instead of using Windows, we use Debian with GNOME, I think GNOME 3, for our student workstations. And then that gets networked into the virtual training area. Everything that we deploy in OpenStack in the public range is successful from the classrooms. And then we also have – In some other tools, tooling we're using right now. This goes back to having that flexibility to kind of innovate on the fly. As we were in the pandemic, one thing we're showing down, we wanted to get people to do distance learning as quickly as possible. So our leadership were willing to [inaudible 00:31:18] let us kind of do things differently by deploying our own Big Blue Button instance in Azure.

So we deployed that in Azure, and then we now have Big Blue Button, which then facilitates like a virtual classroom instance. Now the government have brought on Microsoft teams. So it's less relevant now. But initially when things were first – There's the initial outbreak, we were the only school across the army that was able to adapt and not risk any of our – Some of our functional or kind of secondary courses were cancelled. But all of our primary training has gone unaffected. So far knock on wood, we have not had any outbreaks.

**[00:31:54] JM:** Yeah. I mean, since you mentioned it, when you sat down you said you had just gotten back home from work. And that was a foreign concept to me. That's not something I had heard in many months. Why can't you just do everything remotely?

**[00:32:12] BA:** Yeah. In this particular case, we had a lot of things we can't do remotely. For the army, there are different cohorts that we teach. For officers and warrant officers where we know that they have personal computers, this is one of those things again where we're assuming this is not the normal way the army does things. It doesn't expect people to bring their own equipment. But we know that they all have the money and the resources to have their own internet connections and their own equipment. So they are working from home and conducting class remotely.

Now our enlisted populations, they're in initial entry training. So they're straight out high school. A lot of them may or may not have computers and they don't have their own residents. They live in the barracks. For those populations, we don't have necessarily the ability to give everyone computers in the barracks with internet. Now if there's a contingency where we have to get to that point, we are working to pause yourselves for that of course. And they're working very hard for those to be able to do that. But we can't necessarily have right now all of our enlisted training. Instead of canceling class, we don't have the ability to or to be able to put it on the barracks. But what we do is we spread out the classrooms.

So we have some instructors still come in, but everyone is wearing masks. And then they'll do – Instead of doing it side-by-side, they'll do maybe every three desks. They'll be spread out. They'll be six feet apart and the instructor will be separate. Still be far apart and they're also wearing masks. So that reduces the risk there so that even if there was someone who was annoyingly infected or asymptomatic, there's very low risk that they would infect anyone else.

I'm really proud of command care. They've actually been very proactive in the very beginning and have taken very necessary precautions. And we're in Fort Gordon, we're in Georgia. So we're taking precautions to do the best we can to protect this. The population on base where we have – Unless it's soldiers in the barracks, they can't get out of – They're essentially isolated. And so they can't go anywhere. But then we also don't want to get them infected because they're in close proximity. The Army at Fort Gordon is doing a very good job of maintaining that. And hopefully if all goes well, we'll continue to.

[SPONSOR MESSAGE]

**[00:34:21] JM:** Today's sponsor is Datadog, a monitoring and analytics platform for cloud scale infrastructure and applications. Datadog integrates with more than 400 technologies. So you can track every layer of your complex microservices architecture all in one place. Distributed tracing and APM provide end-to-end visibility into requests wherever they go; across

hosts, containers and service boundaries. With rich dashboards, algorithmic alerts and collaboration tools, Datadog provides your team with the tools that they need to quickly troubleshoot and optimize modern applications. You can see it for yourself and start a 14-day free trial, and Datadog will send you a free T-shirt. Just go to softwareengineeringdaily.com/datadog and learn more. Get that free cozy T-shirt and start your 14-day trial of Datadog. Thank you to Datadog for being a sponsor of the show.

[INTERVIEW CONTINUED]

**[00:35:21] JM:** To revisit the kind of exercises that you are digitizing or formalizing in the Courseware as code practice. You've mentioned this capture the flag term a few times. I think we should actually clarify what that is. What is a capture the flag exercise?

**[00:35:38] BA:** Right. This borrows from kind of the more, the hacker culture. So when I referenced when I was a cadet, there are these – The basic premise of capture the flag from an offensive or defensive sometimes scenario is you have a discreet challenge. They'll be like, "Hey, here's a vulnerable web application, or here's a vulnerable binary. Here's a reversing challenge. Can you take the file? Can you reverse engineer it?" And when you reverse engineer it, you find a token or a string, or some unique string that is identified as a flag and you [inaudible 00:36:11] scoreboard and you get points.

Same thing with the web exploitation. You exploit a database. Some might be a flag hidden in one of the admin's hash or something like that. This is a common practice throughout industry for the capture the flag formats. But then the great thing about the format itself – A challenge with capture the flag is often if you're a new person and you're trying to get into it, capture the flags can be a lot of fun, but they can also be demoralizing. Because usually you'll start with maybe a few easy challenges and it gets really hard really quick. But the format itself allows you to steal a large amount of curriculum into discrete individual skills.

The first time, if you're starting at the very beginning, you might ask someone to list files in a directory. And one of the files might be a flag. This isn't necessarily how basic we start in our curriculum. But as a concept, especially if you're reaching out applying this to more entry-level,

then you might have people change directors. Then you might have them show the IP address. So you start working through commands. And each challenge would be you do the challenge. The students have to work through the problem and figure out how to solve it. So they're building problem solving.

And then when they find the flag, they get instant feedback of, "Oh! You did it right. You did it wrong." So it's very iterative. And then when you get feedback, it builds confidence. And if you are able to design a capture the flag challenge, a series of challenges such that the slope of difficulty is very gradual slope rather than a steep slope, then students are able to build confidence in their own problem solving ability and are much more willing to keep working and keep trying. Because one thing about the information security space is sometimes the different between someone who can in a capture the flag or in some challenge solve something as someone who doesn't. It's not necessarily that they have different skills. Some person might just be willing to spend 10 hours looking at our problem, or a week, or a month and hacking through something until they figure it out.

Part of that is a confidence that if I just keep figuring this out, I'm looking at this, I might be able to figure it out. Capture the flag is, in my mind, a way to help build that self-confidence that, "Hey, I can solve this problem, because I just keep working at it." And then you build it up gradually, gradually, gradually. And then before you know it, people are getting to more complex skills and they're developing more problem solving and they're getting more self-confidence. Soon you'll get into the harder topics.

So when we use that in our Courseware, we have exercises based on lesson plans. So the students will get to try this, but then it also has some things that will go above and beyond the lesson plans so students can kind of spread – Or like they get done with everything or they know everything already in the lesson plan and they can try more hard concepts that might go above and beyond what they've learned. So it helps us identify people who stand up above their peers. It might be able to fast track or go to a certain assignment. Or, "Hey, this person could be good for a specific work role in the operational force because they're good in this area," and things like that.

The capture the flag in my mind for the cyber school and also for across any educational institution I think pairs very well with Courseware as code, because you can tie the Courseware and the capture the flag interface, capture that all in code and then deploy it to some infrastructure cloud, whether that's Azure, Google Cloud, AWS or something like OpenStack that we self-host. Or whatever cloud you want to use. You can pair that all together and package up nicely. And then eventually we don't do it at the school, but I would love as like a personal goal to get this type of concept to a place where someone could off the street pull things off the shelf, pull some Courseware as code, package off a repo, run deploy in a pipeline and then they point it at their cloud and they get a full-featured interface with Courseware, the infrastructure and the gamified interface all packaged together.

We at the school last summer sponsored from a manpower's perspective. So some of us volunteered to support a cyber-patriot camp downtown. We partnered with some local partners in Augusta and to run this for high schoolers for free. So they paid all the bills, because they're the private ones and they have the money and the government can't pay them any money. But then we volunteered ours as military members to help instruct.

So as a part of that, I spent a lot of time on nights and weekends building a capture the flag interface for that where I track all the challenges in code and then build a pipeline to build a zip that could import in a CTFD. And so then I can out of the box just have a really quick capture the flag interface that does that somewhere gradual slope to help students learn. I think this kind of apply both for the government and then also for partnerships in the government and private sector as well.

**[00:40:36] JM:** Do you ever wonder why isn't there more dedicated software I could use to managing this? Does it feel like you've kind of patched together this ad hoc workflow that for some reason there's not some tool out there you can manage it with? I mean, the fact that you have to use Git to manage this Courseware, doesn't that feel a little like the space is immature or there should be some off the shelf thing that should do this?

**[00:41:09] BA:** I think GitLab would agree that our use case is especially for – Their tooling is very unique. I don't know if they have anyone else that uses it quite like we do. At least right now there are others who do use it to track markup languages and so on. But the full feature pipelines and everything, I agree, it's piecing together, gluing together a bunch of technologies to make something. But I know there are – I think the difference, there are people who probably do this type of these things in products. But for them, they're proprietary. So it's going to be very expensive to – They'll come and piece it all together and might even deploy things with pipelines, but they're going to try to sell it for a lot of money, because that's how things work.

I think what I'm really excited about with Courseware as code and kind of applying kind of that [inaudible 00:41:55] open source mindset is, for me, I care a lot about – Service is a kind of a value of mine. So I want to give back to the country and. For me, Courseware as code I think is a tool to kind of reduce the barrier to entry for someone who might be underprivileged or grew up in an environment where they didn't have money or something to pay for a $5,000 course, or something like that where they do have all those things where they have those technologies kind of they do all themselves and they're proprietary in-house and kind of open it up for people to be able to share it.

The school in the past – This is a value of the school has, and sometimes we're able to do it and sometimes we're not. But we would like to be able to share as much as we can. Sometimes you can and sometimes you can't. But certainly, one, we are able to – We try to keep things public and open so that we can give back to the community for that. Our GitLab, we have some things that are private and closed-off, like class materials. But the public's content that's there is actually public. As much as we're able to within government regulations and such with curriculum, if it's not releasable to now. But sometime we're able to get things reviewed to be releasable. I would love to [inaudible 00:43:01] instance right now. But in the past we had our potential market lab push to public repos, but it doesn't happen right just because the review process can sometimes be challenging. So we can't point to the thing right now. But certainly a value we'd like to keep pushing forward.

**[00:43:16] JM:** Tell me more about some of the technical problems that you're facing right now in developing Courseware as code workflows.

**[00:43:24] BA:** For me personally, I am assigned to building an entry-level course for individuals who have a computer science background who will then work in operational jobs for software engineering. For me, I have a computer science background, but then I don't necessarily have the operational experience. But tying together workflows for me to implement courseware as code. In the past, you mentioned how all of everything was kind of seems like they're being together. I would say because of the nature how the school stood up and we're kind of driving a bus as we're building it. A lot of the concept of courseware in code in theory were not always implemented the way we said we wanted to do it.

For me, a challenge has been to actually – As a proof of concept, write something once and then have it apply, be used across everywhere and apply that. Don't repeat yourself principle so we could write the courseware once and then never have to worry about what formatting, or formatting slides or formatting student books or instructor books or lesson plans. That's all TRADOC standards. For just implementing it one.

I think what's also kind of interesting, people find this in government sometimes. The technical challenges are hard, but the bureaucracy can be more difficult. So the government has necessary – It has processes to rotate people through jobs relatively quickly. So you'll go to a sign and you'll stay there for three years and you'll move. So this has a lot of benefits in that, and that if you have a bad leader, a bad leader can't damage an organization. It makes it more resilient. It prevents the army from being reliant on a single individual on a single place, because the army is designed to be modular, because that's the nature of warfare. You have to be able to plug and play as a culture.

But then the challenge then can be that if you build something and you're the subject matter expert on something, then when you leave, you don't want it to fall apart. So for some institutions, they're able to institutionally – Like a private sector can keep people for a long

period of time or they might have the ability to keep a good thing good, a technical – Something that's very technically successful, but keep it there.

But in the army and in government in general, it's sometimes challenging to make something survive when you leave. For me, I think my biggest challenge, one of my goals is to build something such that when I leave, it will survive and have the foundation to not get replaced or just be so difficult that someone else could have taken it up and move.

I want to talk about those pipelines. I've been working on building pipelines, documentation and all those things to make it survive when I leave. So someone else can pick it up and learn it all by themselves. I think, I guess, maybe the ultimate that comes down to is writing things down. It's really easy to do something and keep it on your head if you're the one person who does it. But when you leave, if it's not ran downed, it's not ran downed in a way other people couldn't access it, then it's not really useful and it kind of goes away when you leave. For me, I think it's almost not really a technical challenge. It's an organization challenge of how do you build something so it can survive beyond you when you leave the organization?

**[00:46:25] JM:** So is that more about building the actual engineering workflows and scripts and stuff? Or is a lot of it about how do you document things properly?

**[00:46:35] BA:** I think it's a little bit of both. People said a good code reads itself. But at the same time, documentation is a big part of it. For example, I build a pipeline that essentially it's somewhat convoluted and sometimes when you're a one-man team, you don't use branches. You write it on the primary branch, whether it's master and main, or whatever you call it. And then you piece all these things together and it's your free time when you're borrowing between meetings and so on. And so I have something that takes a directory structure format, build a dictionary out of it and it feeds into templates to build all of these markdown and Asciidoctor and [inaudible 00:47:09] templates and so on.

So when I say like document, then it's writing an actual course in that format to teach people how to use it. I would write all the actual learning steps and outcomes and all the Courseware

content as an example of here's how to use it and then it actually teaches people how to use it. I mean, it's not really innovative or anything like that. It's just taking the time to write things down at an organization that kind of is always moving and growing at a break neck speed.

**[00:47:39] JM:** Do you have your eye on any other tools or processes that could make better productivity gains from where you stand today?

**[00:47:46] BA:** I think right now our biggest challenge is really adaption of what we have more than new things. I think – From a productivity perspective, we have proven that what we have works in some courses. But then scaling it out to all courses is sometimes more challenging. Because once people get established in the way they run, it's harder to get them to change it, especially while you're teaching as you go.

I can't really think of any new technologies we've been looking at moving to. We tend to lean on – Within OpenStack, you can kind of deploy whatever technology you want. So it's the backend. So you can build your middleware inside of OpenStack. So when we need a new thing, if it's open sourced, we can just deploy it and tear it down and so on. So long as it fits the licensing for that of course. But we don't necessarily have is if someone wants to try something, they can try it out. I don't necessarily know if there's necessarily anything where you'd say I really wanted to try this, but I didn't have an opportunity to.

That said, I think adaption, to get people to use what we're using across the board. Many courses do do it, but some just have already been founded. Getting them to adapt it and do it in a good way is sometimes challenging for us.

**[00:48:59] JM:** Can you make use of public cloud yet in the military?

**[00:49:03] BA:** I can't speak to the DoD at large. Hopefully, if people are following this topic, they would be able to see there's the Jedi contract that's got been in the news a lot. I know that's still going through its process. I don't have any visibility on that. So I can't speak to right now. I know my organization, we do have access to our organization has Azure. So we can host

things there. But for us, we use the cost of hosting a significant – If you wanted to build several. We have, for example, right now, we're running around 7,000 compute nodes on average. If you wanted to run 7,000 compute notes in a public cloud or a virtual course rather, it would cost you a lot of money.

We'd take that and we put that in hardware and try to be good to do it to the tax payer's money rather than throwing it all into public cloud that's always on. From that instance, there is access to it, but we try to – When we – For example, our frontend, we host in public cloud. That would be things like our authentication. GitLab itself is not hosted in OpenStack. We host that, in instance, in Azure. So then we take advantage of their backup. But then GitLab deploys things in OpenStack, and OpenStack is all our personal hardware that we manage in-house, if that makes sense. So we kind of try to balance best of both worlds there.

**[00:50:21] JM:** Is there anything else about building software in the military that might surprise people outside of the military?

**[00:50:28] BA:** I think across the board, people who are not in the military would expect that the government runs and might build its own software, things like that. But anyone who's dealt with acquisitions or understand how government works would think that the government probably buys everything. I think people would be surprised to know that is there's a growing culture within the army and across the DoD to apply DevOps and DevSecOps to the government.

Things like instead of purchasing software from someone where the private corporation might keep the rights and the government is tied to that. The government is looking for ways to, "Hey, how do we buy the data rights for the software too when we purchase this and then we have a flexibility to manage it in-house?"

Army features command is also building their own developer. They call it developer factory or something like that where they're trying to apply software development to army problems internally rather than going through an acquisition process. And then we're able to manage. We might buy something off the shelf and then we keep the source code and we buy data right for

the source of code and then we might manage it in-house. So then we have the flexibility to update things.

For people who may be are familiar with the government from the negative side where it's not agile and it's not able to move, it might be surprising to see that there are efforts to move forward, like Defense Digital Services, U.S. Digital Services. They've been around for several years now and they've done a lot of good work. So there's a lot of innovation in the government to help make things better to try to make it more agile and make it use software development principles across the army and across the DoD and even the federal government.

I think for me personally as someone who's worked in the government, it's surprising. I wouldn't say I'm surprised to see that happen and I'm really encouraged by it and I think there's a lot of opportunity if someone says, "Hey, I want to give back to my country. How can do it? And I have technical skills." I think there're a lot of places whether it's USDS or DDS or the Army Cyber School or somewhere in cyber com or under other federal agencies in DoD. There are a lot of people who do great work every day who kind of don't get recognized and people – They kind of know they're there, but there actually is a lot of people who work really hard to innovate and make the culture better. So that's kind of what I would say, is most surprising to me is that it kind of works and it's changing the culture across the army to change the way it does acquisitions and change the way we manage software.

**[00:52:50] JM:** Okay. Well, Ben, it's been amazing talking to you. Anything else you want to share in closing?

**[00:52:56] BA:** I think we covered a lot here. I really thank you for the opportunity, Jeff, to have us on your show and tell our story at the Army Cyber School. I want to thank GitLab for introducing us. And we sure want to thank you for your time.

[END OF INTERVIEW]

**[00:53:15] JM:** Open source tooling is generally preferable to closed source tooling, because with an open source tool, you're going to know what code is running. You're going to know what the community is saying about that code. And you're going to have flexibility. But scaling open source tools is not that easy. You're going to have to spend a lot of time managing and maintaining that open source software. And the alternative is to use closed-source software, which will be scalable, but you won't know exactly what code is running. And you won't have an easy migration path.

Logz.io is a scalable and fully-managed observability platform for monitoring, troubleshooting and security, and it's all based on open source tools like the ELK Stack and Grafana. Logz.io is open source software at the scale that you probably will need if you are a growing company. Sign up for logz.io today by going to logs.io/sedaily and you can get a free t-shirt. You can go to L-O-G-Z.io/sedaily, create an account for logz.io and make a dashboard. Once you make that dashboard, you will get a free logz.io T-shirt.

Thanks for listening to Software Engineering Daily, and I hope you check out logz.io. That's L-O-G-Z.io/sedaily.

[END]