FEDERAL NEWS NETWORK

EXECUTIVE BRIEFING SERIES:

Digital Transformation

Sponsored by GitLab

GitLab

A full DevOps ~~toolchain.~~*

*No assembly required.

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring and security.

Used by more than 100,000 organizations around the globe

SONY CITRIX Worldline NOMURA

Goldman Sachs EQUINIX SIEMENS EA ERICSSON

U.S. AIR FORCE Freddie Mac LOCKHEED MARTIN ING

How to add speed to software deployment in federal DevOps shops

BY TOM TEMIN

As a foundational part of the ongoing modernization of federal IT systems, agency practitioners are rewriting or replacing applications. Some missions require new functionality. Others need new, contemporary code that is easier to maintain. Many agencies want applications to work across hybrid cloud environments, and legacy applications often do not.

Moreover as they modernize applications, IT shops also want the efficiency and regular delivery of functionality that comes with agile development strategies. They're no longer satisfied with elaborate requirements lists and long project plans they turn over to contractors and hope the software comes out right in three or five years.

Plus, agencies want reusable software, developing functional blocks that they can share among components, confident of the interoperability and security of code. And they want the speed that comes from automation of testing and insertion into systems.

But agile development can bring problems of its own. Fast production resulting from sprints or even shorter spins, can produce modules that might be incompatible with existing systems, introduce malfunctions, or get deployed containing faulty logic, while passing quality tests.

It's possible to ensure fast, interoperable and secure development without imposing restrictions on development tools. It's unrealistic to expect everyone in a multi-component department, much less all of its contractors, to use the same tools. So how do agencies get there?

Federal News Network brought together experts from GitLab and two federal components to discuss these issues and take the pulse of enterprise development efforts in the federal government.

PANEL OF EXPERTS

Todd Barr, Chief Marketing Officer, GitLab

John Jeremiah, Enterprise DevOps Evangelist and Product Marketing Leader, GitLab

La'Naia J. Jones, Deputy Chief Information Officer of the Intelligence Community, Office of the Director of National Intelligence

Frank Konieczny, Chief Technology Officer, U.S. Air Force

Architecture and applications

The Air Force provides a case in point for the challenges of modernization. Chief Technology Officer Frank Konieczny said, "We have the problem of migrating three thousand applications" to new code. Why? For better security, lower maintenance costs, and cloud deployment. "The question is, how can I quickly do this?" The Air Force wanted a strategy to avoid traditional systems integration acquisitions and waterfall development approaches. He said it opted to use its own air staff to do development in the DevOps model, instead. The highest priorities among the thousands are the weapons systems applications.

He cited a project to optimize the positioning, timing and loading of aerial refueling tankers – and get away from manual white boarding – as an example of a software effort the Air Force is trying to get done in an agile and fast way.

Konieczny's organization typifies the need to modernize code in a way that accurately reproduces vital functions but allows the opportunity for improvement and new functionality.

In the intelligence community, La'Naia Jones, deputy chief information officer, names reference architectures as foundational to development efforts for a variety of domains. Her office, she said, "shepherds" component ODNI offices "to ensure that [everyone's] tools, solutions, and applications are interoperable and that they are able to be leveraged by multiple agencies."

The reference architectures apply to functions such as collaboration – email, chat, application sharing, and file sharing. Cybersecurity, Jones said, also has a reference architecture. The ODNI wants to ensure that cybersecurity is a foundational requirement in all software development. She called that approach the "SecDevOps" model. A third reference architecture covers artificial intelligence and machine learning applications to enhance the work of analysts.

Thus, if agencies want to spread development across multiple teams that are close to users and mission, they need to ensure interoperability among the resulting software modules. That brings up the question of tools used in the coding itself.

Many tools in the box

Jones says the ODNI takes a tool-agnostic approach, but with a caveat.

"We're trying to make [development] flexible enough and agile enough to take in new technology so we aren't stagnant," Jones said. Rather than trying to have all groups use the same tools, the OCIO makes sure tools are used to meet what she called the "foundational" principles of the reference architectures.

She said that one foundation of the architectures is the hybrid cloud environment in which the intelligence community operates. Some of the development tools come with the commercial clouds themselves.

At the same time, she said, her office is partnering with component agencies, trying to inventory what applications and tools have been created both in the unclassified and classified-secret environments.

"We're looking at what are common tools and application sets we can leverage … not just for efficiencies, but also so we can have a common understanding and work together," Jones said. She reiterated a main concern, namely building in security during development rather than trying to retrofit it.

John Jeremiah, enterprise DevOps evangelist and product marketing leader at GitLab, called it "giving development teams buoys but not barriers," in which to navigate their projects. He said it's more important to integrate tool capabilities – such as within GitLab – than to try and get everyone to use the same tools.

For Konieczny, when it comes to development tools, "We look at the toolset and we determine, based on the NIST [National Institute of Standards and Technology] standards, and which tools satisfy the NIST controls" for cybersecurity. That ensures the resulting code meets the federal risk management framework, which in turn leads to the authority to operate the software.

Konieczny said right now the Air Force is evaluating more than a dozen new development toolsets to replace older ones. He added, development tools have counterparts in the resulting production code to make sure "the tools I thought were working correctly actually did work."

Another issue both Konieczny and Jones noted is how to develop and deploy applications in lower security environments that can be migrated into higher security environments.

Accelerating development

Jeremiah noted the need for speedy development exists across the government. "Everyone is struggling with the same challenge," he said. "How do we accelerate software development? How do we resolve and eliminate the barriers and the obstacles that slow us down?"

The GitLab platform is an example of the type of product that can help harness agile (or waterfall) software processes.

Jeremiah said it encompasses source code control management, project management, testing, automation of continuous integration, and maintaining security all the way to production. "It's basically a single application for lifecycle management of software development and deployment," he said.

The idea, Jeremiah said, is to provide a way for the security, operations, and quality control teams to participate early in the development lifecycle. That avoids the serial – and slow – process of "throwing things over the wall."

Aiding interoperability when working in classified situations are features that allow low side developer teams to export their entire projects under the GitLab application, letting development teams preserve the state of their project at any point in time, including coders' conversations and code changes. It's all archived and compressed, so it can be shared with subsequent developers in the high side.

Jones said all of the background information provides for future collaboration within teams repurposing a particular piece of software. "By taking all of the comments, all of the background information, even a conversation about what was tested – that can be leveraged and used by developers and engineers at some other place, some other time," she said.

Todd Barr, the chief marketing officer at GitLab, said if DevOps implies development and operations getting together, the term DevSecOps adds security. He said networking teams are also joining the early phases of development. That's important as applications are increasingly deployed in hybrid cloud environments, which brings in more complex networking requirements.

"Having those audiences collaborating together as fast as possible gets you the speed and the cycle time you're looking for," Barr said.

Konieczny said that for the Air Force speed equals functionality. "It's how fast you can get a function out that means something to the airmen who have to use it, not so much that I'm doing X-number of releases per day."

Faster deployment

That idea, too, is gaining currency across government and industry, Barr said. Commercial and government customers are thinking about "how long does it take to get from the concept to deployed in the field. There aren't many tools that can help you measure that."

Barr added, "Cycle analytics is a unique concept. In the context of DevOps, what does that really mean? Not how many developments, but how fast do we get to the field."

In the deployment context, Jones made a distinction between new applications and the far more complex task of re-coding existing functions or building new ones, then inserting them into existing enterprise systems. She said it's important to avoid adding or fixing one thing, only to have it cause a break somewhere else.

"Your goal is to keep your operational system as little disrupted as possible," Jones said. She noted that once a developmental piece of software ingests operational data – it's now operational. Automatic security auditing and documentation is essential in such a situation.

Jeremiah pointed out that a tool like that offered by GitLab can not only monitor the performance of newly inserted modules but automate the fix, "and tell you about it after it's fixed."

He pointed out that with automated testing, deployment and remediation, development systems are also, in effect, production systems taking time out of the coding-to-deployment cycle.

"The key to achieving speed to mission and velocity," Jeremiah said, "is what I would call a pipeline. That's a set of automated tests, automated tasks" that keep software and documentation moving, all managed by the tool chain.

A final thought. Panelists agreed on the value of open source software components, so long as they are tested for security and interoperability with the same rigor as any other code. That's because they save the time needed to code in the first place. Jones said the challenge with open source is making sure the organization is using the latest version, especially in terms of security updates. She said she'd prefer an automated way of monitoring open source to make sure it incorporates the latest security technology, rather than having to spend time on research.

Tom Temin is a freelance writer with 40 years in business-to-business journalism. E-mail him at tom@tomtemin.com