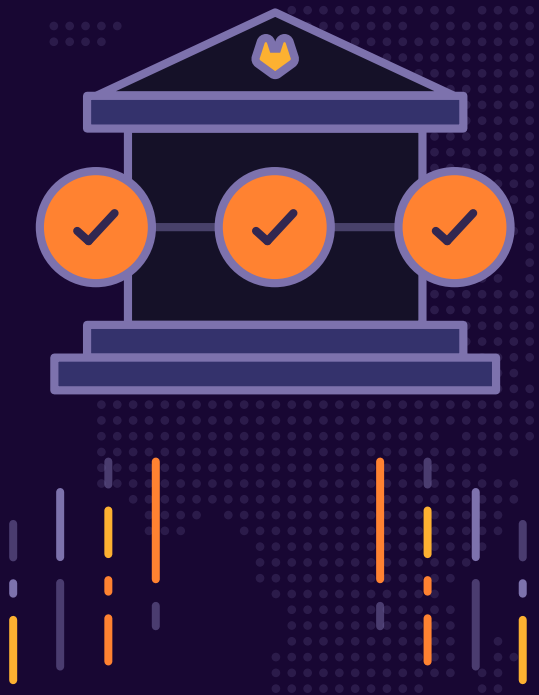
Modernizing Government IT through DevSecOps

A Step-by-Step Guide



GitLab

Executive Summary

Leaders of today's public sector agencies and regulated industry organizations are eagerly seeking to shrug off the legacy systems, infrastructure, processes, and policies that prevent them from delivering programs, projects, and applications that support their missions. They know they must also meet the challenge of providing exceptional experiences to their constituents and customers. Unfortunately, the journey to modernization is even more daunting and complex for government than it is for commercial entities. In addition to overcoming similar technology, cultural, and process-oriented challenges, government agencies, and heavily regulated industries must also work within security, compliance, regulatory, and procurement constraints that threaten to derail even the best-laid plans.

The objective of this whitepaper is to help government leaders:

- Identify the top challenges public sector enterprise organizations and regulated industries face when approaching modernization efforts
- Learn how commercially followed best practices can be applied to regulated enterprises
- Consider step-by-step actions to lay a solid, scalable foundation for transformation
- Envision the future state of the IT organization and workforce

Lastly, this paper will provide actionable recommendations leaders should take to spark sustainable transformation through the lens of DevSecOps within an organization.

Recommendations

1. **Perform an honest assessment**. Enlist help to ascertain where on the DevOps maturity spectrum your organization actually is in order to determine the best strategy for moving forward. Create the ability to measure performance. Build the capacity to measure throughput and value streams.
2. **Deliberately change your culture**. Allow teams to fail, promote experimentation, empower people to find the right ways to work together, provide management instruction, and make the 'right' path the easy path.
3. **Transition from waterfall (or 'scrummerfall') to full-fledged Agile methodology**. Changing your processes, tactics, techniques, procedures, and tools will impact your policies and your ability to accelerate delivering on your mission.
4. **Rethink your acquisitions strategy**. Create next-generation contracts and projects that are more mission-based and less prescriptive to leverage commercial best practices.
5. **Build in economies of scale and pick the right system to manage it**. Start with aggregating all of your code into one software configuration management system and promote that system to all your stakeholders so that it becomes the core for all development, operations, and security activity.
6. **Move to a continuous integration (CI)/continuous delivery (CD) pipeline posture**. Begin by aggregating on a single platform for the entire DevOps lifecycle to reduce toolchain complexity, integration, management, and maintenance, and to enable end-to-end visibility for the entire team.
7. **Implement a software factory model**. Build out software capabilities that enable faster, better, more secure applications with a fully functional assembly line that is efficient, easy to manage, and able to quickly build, test, and deliver applications without the waste and overhead of managing dozens of disparate tools and bespoke integrations.
8. **Build out an ecosystem and automation incrementally**. Be cognizant of what policies, processes, and skill sets your enterprise has today and evolve your automation as these capabilities progress.

9. **Make visible changing risks as software evolves to address security continuously, not just 'shift to the left'**. From a process perspective, enable your network's authorizing official to think through risks and make decisions upfront and implement governance and compliance to help ensure capability delivery speed.
10. **Adopt technology that allows real-time, centralized communication and collaboration**. Break down silos and eliminate sequential and friction-fraught development, ops, and security handoffs to lay the groundwork for better, faster, and less painful application delivery.
11. **Drive leadership-led change based on the mission**. Connect goals and objectives to how these changes will impact and enable your specific mission (e.g., deploying a capability faster, security, scalability, etc.).
12. **Reskill your workforce**. Employ non-traditional methods like free online courses, networking events, and collaborating with other government agencies and commercial peers to learn best practices that can be applied to your environment.
13. **Build a center of excellence (CoE) and implement programmatic modernization**. Leverage existing agency CoEs to shortcut your modernization journey.

Top Challenges Agencies Face When Modernizing

As public sector agencies endeavor to migrate from a 'wild west' development environment featuring multiple orchestration tools and daisy-chained configurations to more mature, streamlined platforms that teams across the organization or multi-tenant teams can actually leverage efficiently and effectively, they must confront the same demands as their counterparts in financial services, healthcare, and commercial enterprises. They need to go faster, meet compliance requirements, and demonstrate to auditors that they're doing what they said they would do and be able to prove it.

The typical challenges commercial enterprises encounter when attempting to modernize infrastructure and delivery processes are well understood; however, the additional unique constraints public sector agencies and regulated industry organizations face make modernization even more convoluted and problematic. Increased imperatives for security, compliance, and legal regulations as well as acquisition laws and policies further complicate an already ambitious—and at times, painful—undertaking, requiring a thoughtful, balanced strategic and tactical leadership approach.

Generally speaking, all enterprise organizations face the following obstacles to modernization:

- Addressing the **need for fundamental organizational transformation**, not just technology automation
- **Pioneering** new processes, technologies, and approaches
- Overcoming **cultural reluctance** to change
- Galvanizing teams for **delivering code in an organized manner**
- Expanding and **enabling capabilities out to the entire functional ecosystem** of users
- Building out a **secure DevOps infrastructure and pipeline**, even when users are offline or on the network's edge
- **Shifting security 'left'** and incorporating it into the development process
- **Managing legal, regulatory, and compliance controls** for authorization while expediting software deployment
- **Accelerating speed to delivery** of projects when antiquated acquisition processes and regulations do not support Agile methodology

Organizational Transformation

Perhaps the most imperative aspect of digital modernization is not the technology transformation itself, but the transformation of the organization's culture. In fact, it is arguably the most fundamental step leaders can take to ensure the long-lasting changes they seek to implement. Technology and automation are essential—but only after executing and documenting process changes, introducing and adopting those changes in day-to-day operations in a sensitive and intelligent manner, and creating an environment that encourages and incentivizes support from personnel within the organization. Technology restructuring without cultural transformation is a futile exercise leading to wasted investment, internal recalcitrance, further dysfunction, and ultimately, failure of long-term adoption and change.

Being First

Being first in your agency or organization to begin modernization initiatives can be an education unto itself. From getting everyone to understand the value and benefits of the initiative, changing the processes and mindsets of key stakeholders and personnel in the organization, selecting and implementing the right platforms and tools that will enable adaptation and growth over time, to demonstrating the changes to auditors and authorization officials, nothing happens easily or overnight. Pioneering radical change within an organization is never frictionless and is usually replete with uncertainty and reluctance to change.

Cultural Reticence

One of the key challenges modernization leaders see when undertaking digital transformation initiatives is a deep-rooted cultural aversion to change or to investing time and effort into new methodologies. They find that their people are mired in non-Agile, waterfall-like program management practices or are dabbling with Agile but really haven't committed to the approach (practices known as 'scrummerfall' or 'water-scrum-fall'). Some have adopted the Scaled Agile Framework (SAFe) but have found that the complexity of the framework can lead to lack of genuine collaboration and self-organization (and therefore unnecessary, brake-pumping hierarchy), loss of the customer voice, and long feedback loops.

From an operations perspective, these teams are dealing with outdated monitoring, considerable segregation of duties, and antiquated ITIL practices. Communication between teams from different companies (especially among contractors that may be competitors) is usually siloed and disjointed, lacking transparency.

Organized Code Delivery

In organizations that have allowed homegrown application development tool adoption—particularly where developers have selected their favorite tools amidst innumerable choices or in those groups that have merged or integrated multiple developer and operations teams—delivering quality code on time can often be more art than science. Frankenstein-esque amalgamations of the toolchain are notorious for not scaling well, being hard to maintain and even harder to integrate, and preventing a simplified, predictable software development lifecycle (SDLC) process. The lack of organized, repeatable processes delays delivery, impairs quality, increases vulnerabilities, adds re-work, and creates compliance nightmares.

Delivering Capabilities to the Edge

Expanding delivery system capability out to the far reaches of an organization's broader community—not just to users and stakeholders who are geographically nearby, but to the functional edge of citizens, customers, warfighters, and other constituents globally—creates further technical and logistical challenges. Democratizing the ability for functional teams to develop and deliver applications in any environment while enabling end-to-end visibility into software analytics, quality, and security issues can be seen as an elusive pipedream in many organizations.

Securing the DevOps Infrastructure and Pipeline

Even in cases where organizations have been able to bring capabilities to the functional edge, they still face the obstacle of ensuring that users who are disconnected from the internet experience secure and reliable availability. For instance, just how do you build a secure DevOps infrastructure and pipeline that meets the unique requirements of the battlefield environment, especially when it may be disconnected from the internet?

Incorporating Compliance into the Development Process

Government agencies are often encumbered by the authorization to operate (ATO) process that begins at the completion of the development cycle. The painfully extended time to ATO is further exacerbated when the development and operations teams are still using waterfall or scrummer-fall practices that place security at the end of the process, leaving vulnerabilities and expensive security fix remediation to be completed as unplanned and unscheduled work.

Managing Controls vs. Delivery Speed

Agencies and organizations face constant compliance, legal, and regulatory requirements as well as additional enterprise-level controls that need to be met. Faced with the spectres of legacy systems, outdated processes, sequential development practices, and extended authorization timelines, many IT leaders are engaged in a formidable battle to deliver the capabilities and results their mission demands. How can they accelerate their pipelines? What can they do to move to ongoing authorization that is integral to their pipelines? How do they provide audit trails of every change to their code?

Overcoming Sluggish Acquisition

Current antiquated acquisition practices—built on the waterfall software delivery model and put in place decades ago—are not Agile and were never designed to support the modern approaches required for delivering projects at speed. Outdated vestiges like the Contract Data Requirements List (CDRL) and Data Item Definitions (DIDs) perpetuate the core problem: acquisitions are too prescriptive, telling contractors how to do things instead of what needs to be accomplished (the mission or objective) and most don't break down work into smaller sprints that would enable the agility to deliver the best result, on budget and on-time. Legacy acquisition approaches can result in bloated deliverables that don't actually bring the best solutions to the mission.

Leveraging Commercial Best Practices

By identifying, validating, and adapting the lessons learned from commercial enterprises, particularly those in regulated industries that face similar challenges to government agencies, public sector organizations can avoid some of the pitfalls inherent to modernization and the transformation journey and potentially leapfrog their efforts to achieve their goals without unnecessary implementations along the way for incremental technology and practices that can complicate and derail progress.

Some of the best transformation practices by successful enterprises include:

Embracing Full DevOps Adoption

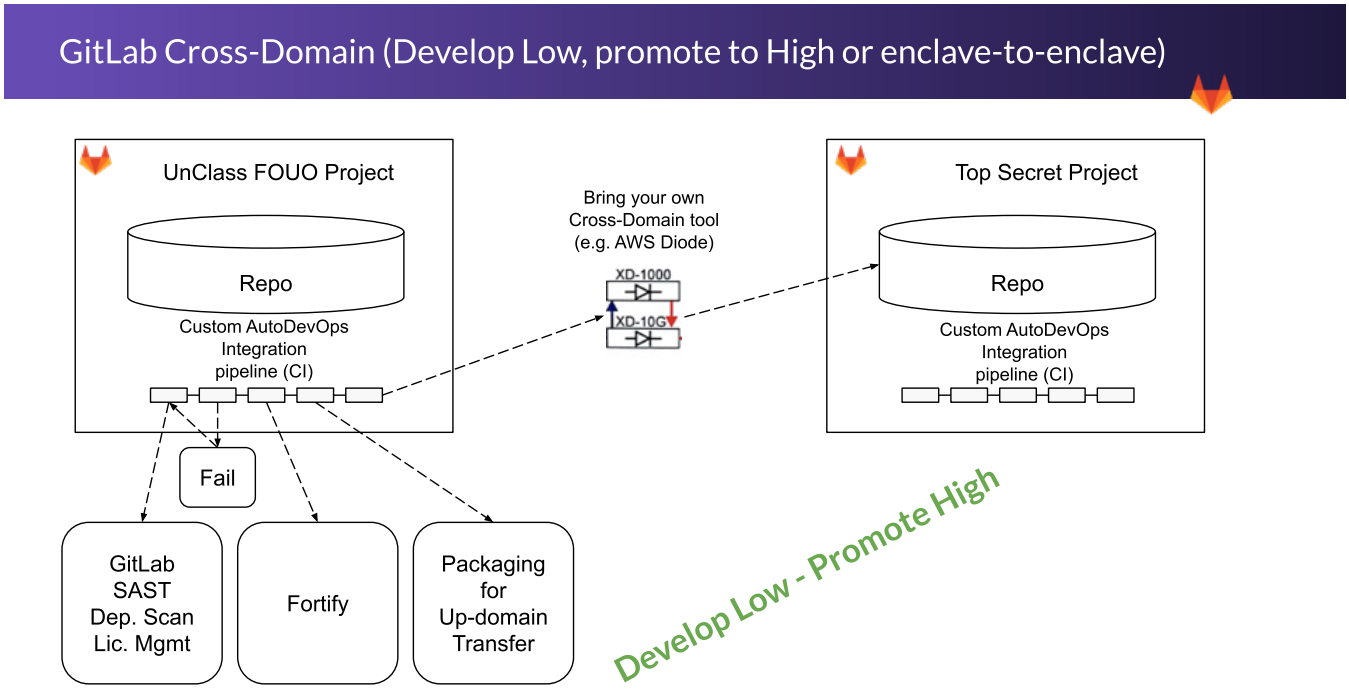
Face it...scrummerfall isn't really Agile and it's only marginally better than waterfall. To realize the benefits of faster, better, more secure applications, enterprise leaders must shift teams to fully adopting the culture of DevOps and embracing the spirit of continuous improvement, supported by the right technology. By realigning teams and their work—breaking down silos, eliminating handoffs, and incorporating security into the development process—enterprises empower those teams to get the needed capabilities out the door quickly.

Building an Ecosystem, Tooling & Automation

One of the most transformative opportunities open to IT leaders once a mature, modernized infrastructure is in place is to build out an ecosystem to support a DevSecOps environment. They can then implement tooling and automation to enable far-flung teams and users to share, collaborate, and function efficiently without requiring them to be co-located in a NOC or in the same geographical region.

Enabling Asynchronous Collaboration

When team members are geographically dispersed—across time zones or at the network's edge—asynchronous collaboration capabilities allow them to contribute to projects without creating bottlenecks for others waiting for handoffs across the DevSecOps team. Once reconnected and back online, code contributions can be easily and quickly merged back into the pipeline to get it into production, resulting in faster time to value for the mission. The team can leverage capabilities to replicate repositories and pipelines on a high-side environment that is mirrored daily to enable the same work asynchronously as when internet access is available.



Example architecture of developing on the Low side (or on one enclave) and promoting to the High side or another enclave, complete with ALL conversations and artifacts.

[1]

Designing Capability-Focused Pipelines

Start with the end in mind when designing your pipelines. What capability is the organization attempting to build? Design your DevSecOps pipelines to focus on the type of capabilities and user experience you want to achieve. Just as a commercial entity that creates web apps should design processes and implement tools that support creating web apps, agencies should assess the end goal and build their pipelines to reflect that mission.

Step-by-Step Considerations for Laying the Foundation for a Scalable Enterprise

Make an Assessment

Perform an honest assessment. Enlist help to ascertain where your organization is in order to determine your strategy for moving forward. Depending upon your maturity level, where you want to go will dictate milestones along the way and what you need to do to start your journey. Some organizations attempt self-assessment (see the GitLab DevSecOps Methodology Assessment); however, you may find that professional organizations like DevOps Research and Assessment (DORA) can provide an assessment to help you understand your maturity level.

In addition, you will want to create the ability to measure performance over time as you progress. Ensure that you build the capacity to measure your throughput, value streams, and workflows.

Change the Culture

Deliberately change your culture. Allow teams to fail, promote experimentation, empower people to find the right ways to work together, provide management instruction, and make the 'right' path for your people the easy path to follow. Changing the culture is a very conscious and deliberate undertaking, requiring top-down, leadership sponsorship on larger initiatives that push the enterprise forward (so be careful not to waste time on small, isolated projects with low impact).

Provide teams the ability to fail and to learn that failures can actually often be successes. Budget time to promote experimentation and empower people and teams to discover how they fit, what tools support team objectives, the right communication structure for their teams, and how they are going to work properly to meet the goals of speed and quality.

Transition to Agile

Transition from waterfall (or 'scrummerfall') to Agile. Transitioning from a more traditional waterfall to an Agile approach should be your first step in driving transformation. Changing your processes, tactics, techniques, procedures, and tools will impact your policies and thus your ability to accelerate delivering on your mission.

Rethink Your Acquisition Process

Reevaluate and rework your acquisitions strategy. Create next-generation contracts and projects that are more mission-based and less prescriptive to leverage commercial best practices. Start thinking about what a next-generation type of contract or organizational construct would look like. Don't tell your contractors and vendors **how** to do their job (after all, their expertise is why you hired them in the first place), tell them **what** you need to accomplish and allow them to bring you innovations from their experience and commercial practices. Build SMART (Specific, Measurable, Achievable, Relevant, and Time-Bound) service level agreements (SLAs) with all vendors and integrators that support relationships between the government customer and the integrators and are focused on the mission, development cycles, iterative delivery, and cost management.

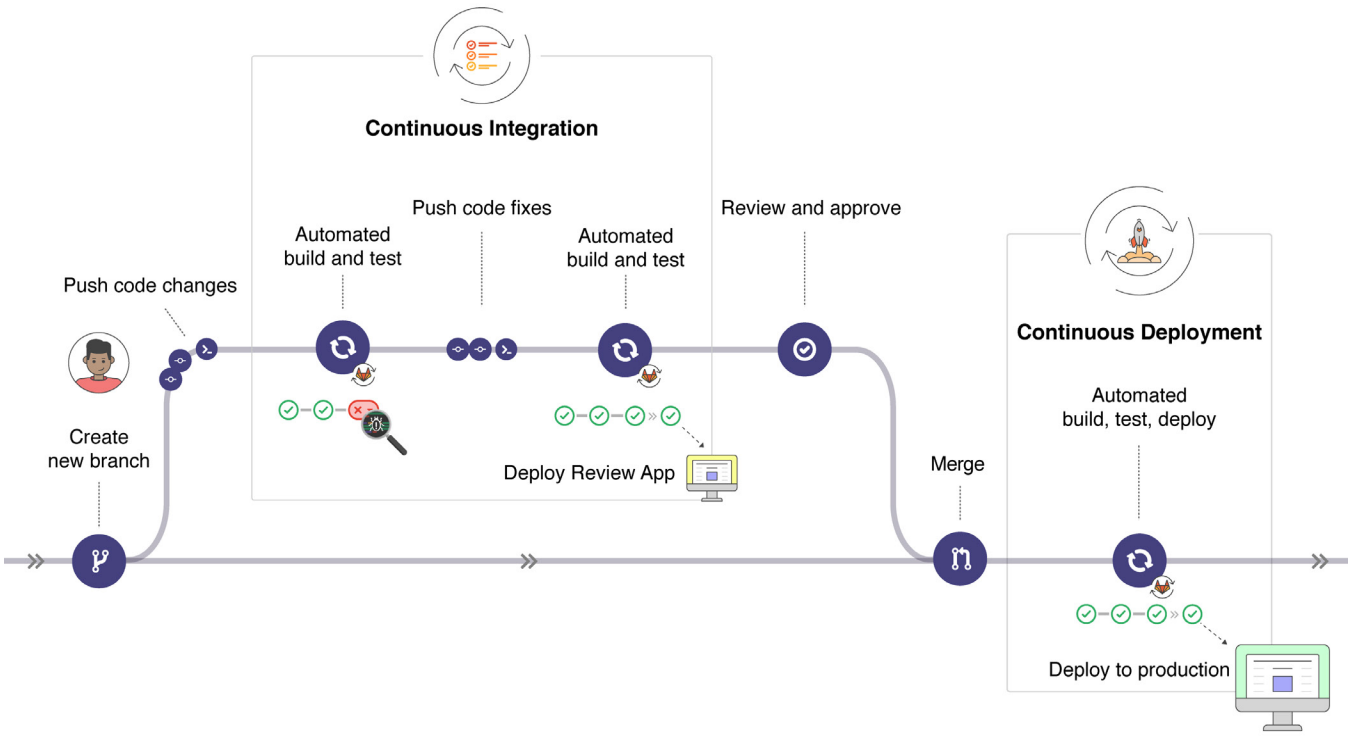
Procurement management can also be handled more effectively by using an Agile approach and the right technology. For example, procurement representatives can use an end-to-end platform to track changes to project scope and schedule impact, contracting officer's representatives can modify contracts, and contracting officer's technical representatives can ensure deliverables are on-time and meet specifications—all in the same place the development, operations, and security teams are working.

Pick the Right Development Platform

Build in economies of scale and pick the right system to manage it. Start with aggregating all of your code into one software configuration management system and promote that system to your entire stakeholder base so that it becomes the nerve center for all development, operations, and security activity. This platform should become the core for a wide variety of collaborative activities like standing up internal training websites, knowledge transfer, project management, and rigorous version control.

Move to a CI/CD Pipeline

Move to a continuous integration (CI)/continuous delivery (CD) pipeline posture. Begin by aggregating on a single platform for the entire DevSecOps lifecycle to reduce toolchain complexity, integration, management, and maintenance, and to enable end-to-end visibility for the entire team. You can then start to move to a CI/CD pipeline mindset. If you have a number of different teams using a plethora of artifact repositories, start to aggregate those repositories to a common set of dependency management tools. Put in place a supply chain for your developers and your operation teams that includes a registry that is transparent for all—not just individual teams. At that point, you can make adjustments to get to that single DevSecOps lifecycle application based on what works best for your overall team and objectives, gradually phasing out tools that are redundant or that do not support end-to-end visibility for CI/CD.



The GitLab CI/CD workflow is able to track the entire process, without the need for an external tool to deliver your software.

Implement a Software Factory Model

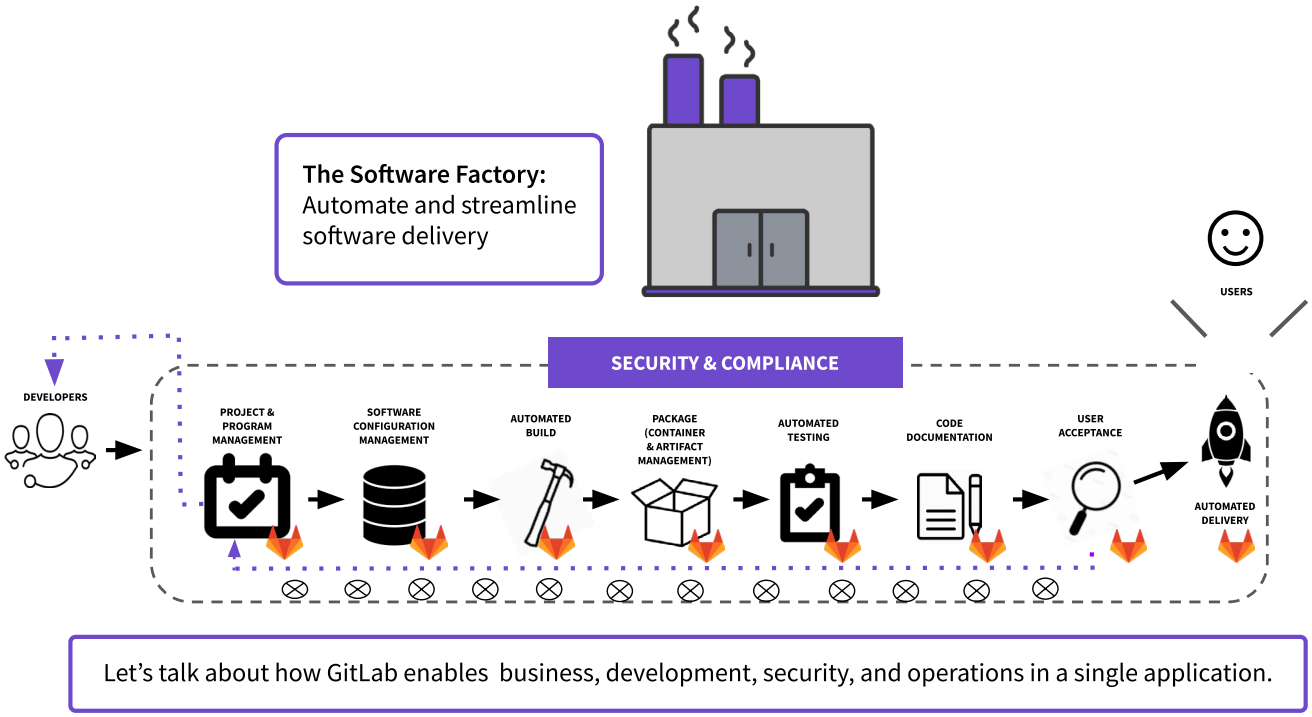
Put in place an integrated, out-of-the-box, modern software development factory. An assembly line in the form of a complete DevSecOps platform delivered as a single application is an efficient, easy-to-manage path to quickly build, test, and deliver applications without the waste and overhead of managing dozens of disparate tools and custom integrations. An integrated software factory also provides a single source of truth for centralized, asynchronous collaboration across a wide variety of roles and is the key to meeting compliance and demonstrating auditing trails. The factory's consolidated, end-to-end view of code quality and security enables remediation during the development process, which in turn, enables better quality code and faster delivery as well as fewer development delays and more on-time releases due to a reduction of rework. An effective software factory has one interface, one user model, and one data model for the entire DevSecOps lifecycle.

Criteria for an effective software development factory:

The public sector needs to be able to deliver secure, high-quality code efficiently in a multi-cloud, cross-enclave environment...all in a way that meets compliance requirements.

The software factory that meets these requirements for the public sector must:

- ☐ Give delivery teams the ability to capture, discuss, prioritize, and define new requirements and use cases
- ☐ Allow coordination, sharing, and collaboration across the entire software development team
- ☐ Enable peer reviews and rigorous approvals for code changes and clearly document and track them to demonstrate compliance
- ☐ Automate continuous integration development tasks that are completed for every code change along with automated testing and security scanning incorporated into the development process
- ☐ Manage and track the code and libraries of the application through testing, validation, and deployment
- ☐ Automate the tasks and steps required to take the application from the development and package stages through the deploy and configure stages of delivery
- ☐ Support on-demand, dynamic test environments for testing by individual developers and teams leveraging containers, containerization, and the cloud
- ☐ Enable the flexibility to ship code quickly, iteratively, and incrementally while actively managing and mitigating risks
- ☐ Provide feedback and actionable insight from the application in production so developers can detect issues, take action, and continuously improve the application



The optimal enterprise software factory enables automated business, development, operations, and security workflows—all in a single application.

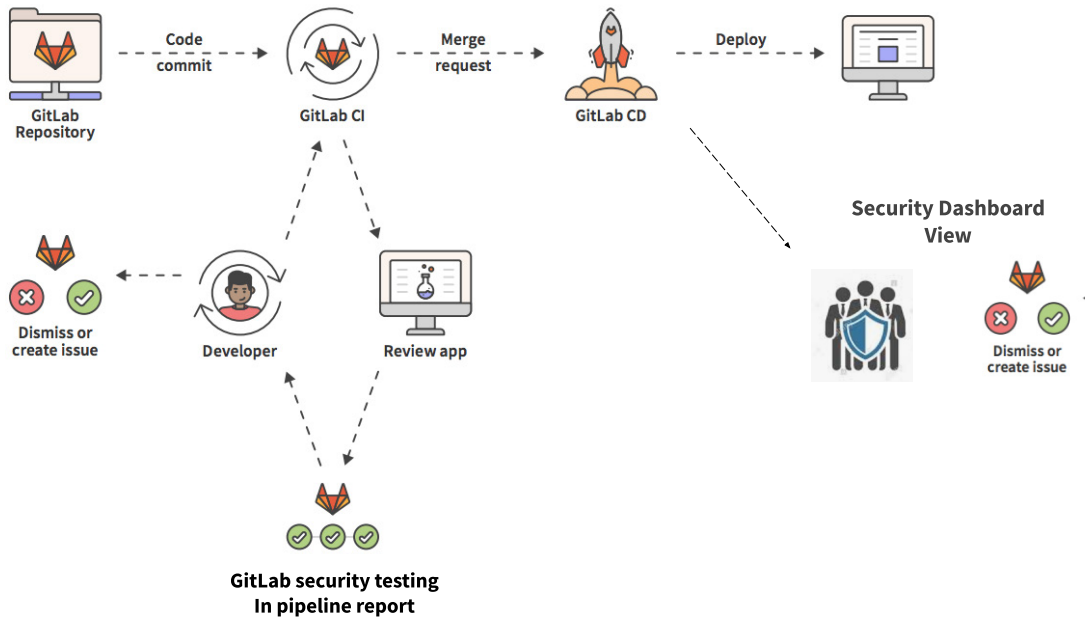
Build Out Your Ecosystem & Automation

Build out your ecosystem and automation incrementally. Be cognizant of what policies, processes, and skill sets your enterprise has today and evolve your automation as these capabilities progress. As you build your DevOps pipelines, remain tightly focused on the type of capabilities you are introducing while ensuring the entire enterprise follows basic Agile principles.

As you gain momentum towards maturity by demonstrating consistent and reliable configuration management, DevSecOps processes, and Agile acquisitions, you will be able to push your capabilities faster and your speed of application delivery will increase. You can then roll out automation in 'subtle chunks' to accommodate the enterprise's policies and processes as well as the skillsets of your team.

Shift Security ‘Left’...and Address It Continuously

Make visible changing risks as software evolves to address security continuously, not just ‘shift to the left’. From a process perspective, enable your network’s authorizing official to think through risks and make decisions upfront and so that you can collaboratively implement governance and compliance to help ensure capability delivery speed. Instituting thresholds, safeguards, and vulnerability remediation within your pipeline at the developer level will not only increase speed to delivery but also radically reduce time to achieve ATO.



2

Empowering your development team to identify and address vulnerabilities before the security team ever sees them reduces risks and allows the security team to focus on exceptions.

Empower Communication

Adopt technology that allows real-time, centralized communication and collaboration. Actively break down silos and eliminate sequential and friction-fraught development, ops, and security handoffs to lay the groundwork for better, faster, and less painful application delivery. Focus on your communications technology: change from using static, inefficient email as your primary communication tool and start to leverage existing technology or adopt new technology that enables real-time, mission-based collaboration to transcend silos of people, projects, and contracts.

Lead the Change

Drive leadership-led change based on the mission. Connect goals and objectives to how these changes will impact and enable your specific mission (e.g., getting a capability out faster, security, scalability, etc.). Recognize that getting people to let go of 'scrummerfall' requires leadership to provide different goals and objectives that help them see the **why** for going faster and evolving.

» Focus on what capability you want to deliver at an enterprise level (i.e. "we need to get a capability out faster", "we need to be more secure", "we need to be able to scale development") to establish the 'why'.

» As your team starts to implement changes and apply some of their learnings, they will make mistakes. You can tell them about what it looks like to go fast, but it requires a maturity gained through experience to get there in reality.

» Create management instruction. This could look like policy but it is also a CIO set of instructions to help incentivize people and teams. Start with just an Agile policy or management instruction. Then iterate and create the next management instruction, providing a carrot for development teams as well as operators if they follow specific best practices (e.g., if you have a release readiness review as a governance gate before you can deploy the production, encourage teams to start to adhere to continuous integration, automated testing, a certain number of deployments per every two weeks, etc. If they follow the best practice, teams can then have the release readiness review gate completely removed, giving them the ability to deploy at-will).

» Reuse tools to help spur along teams and prepare them to eliminate the governance gates (another vestige of waterfall), empowering those teams to take on the challenge of meeting DevOps 101 criteria from a continuous integration perspective.

Focus on People

Understand that change is very scary for most people. There will be concern among your teams that the changes—including new technology and automation—will take over their jobs and they will be displaced. It is critical that they understand that the reality is exactly the opposite. Instead, they will now be freed up to do higher-level, more interesting work.

Most teams use this opportunity to “re-mission” or “rebrand” their workforce, leveraging team members’ skillsets in fresh, new areas, which leads some internal critics to become strong champions for your changes because of what they learn. These champions then continue to advance the transformation as they contribute their experience and learnings to influence processes and policies in a thoughtful, less risky manner than outside technologists or those without hands-on experience ever could.

Train for Today...and Tomorrow

Reskill your workforce. The traditional model of Workforce Transformation (“Reskill, Retool or Redeploy”) is crucial to supporting your modernization goals. However, non-traditional methods of reskilling are likely your best approach for preparing your teams to manage the new environment. Even if budgets are tight or non-existent, you can employ methods like free online courses (e.g., Coursera, edX, or other MOOCs), networking events (like DevOps MeetUps®), industry working groups, government associations, and collaboration with other government agencies and commercial peers to learn best practices that can be applied to your environment. Include a variety of general training, running from a couple of hours or two of listening to full hands-on workshops that cover things like Agile and Lean, to DevOps principles like understanding how to build a pipeline, fundamentals of infrastructure-as-code, or using common development tools.

This is an excellent opportunity for your team members to cross-pollinate ideas both with peer DevSecOps experts in the Federal and commercial spaces and to build new networks. Those relationships are invaluable in softening the transition and in helping to coalesce new skills. In addition, you can foster networking among your teams with other government colleagues who have mature capabilities and can help your teams navigate new possibilities in a safe and secure manner while evangelizing DevOps transformation. For example, the DevOps Federal Interagency Council (DFIC) is a networking group aimed at promoting DevOps practices among Feds and industry. Networking plays a dual role in helping to spread the word and in obtaining workforce buy-in.

Get Compliance Right

Ensuring compliance and accelerating their ability to achieve ATOs significantly is a priority for many agencies. Some agencies have reduced time to obtain ATO from six months to 30 days and have established key principles and best practices that teams, groups, and other agencies can adopt:

» Treat infrastructure-as-code and pick the right tools to manage that level of automation. Set up pipelines that parallel your master pipeline and be able to show that evidence to your auditors and regulatory bodies.

» Institute reusable building blocks that can be leveraged across the enterprise. Work with other agencies and components to share code repositories so that they can leverage them as well for hardening operating systems and pervasive applications that have benchmarks (e.g. 'harden against RedHat or Windows' or specific databases like Postgres).

» Adopt pipeline-as-code so that the entire change management process is documented and is a pipeline unto itself, which then provides regulatory bodies complete auditability. Move from a risk-based, task-driven or tasklist-driven approach to an outcome-based approach to see the results at any point in time and on-demand.

» Refer to NIST Special Publication 800-171A (June 2018) "*Assessing Security Requirements for Controlled Unclassified Information*" for guidance to the path to preparing for compliance audits.

» Provide development and operation teams a parameterized dynamo file to detail locations of endpoints to enable them to move forward without a heavy labor lift to start and automate a project. This file will create a more advanced method of reusability as well as continuous improvement of that pipeline and of your tooling.

» In terms of governance, enable full situational awareness and understanding of what has already been done, how it has been deployed, what the risks are, and how that risk fits into the programmatic portfolio. It is important to have end-to-end, real-time visibility that allows you to see your data and pipelines at all times.

» Automate and document as much as you can to ensure you have visibility into how each piece of code—from 'birth' to deployment and then to 'death'—can be seen in your enterprise. This information should not reside in a spreadsheet that tells you where assets are or what release you're on.

Following these steps should result in a continuous authorization enabled by continuous monitoring, continuous deployment, and continuous integration (as outlined in the NIST cybersecurity framework and in the DoD risk management framework).

Leverage a Center of Excellence (CoE)

Take advantage of your agency's—or GSA's—CoE program to shortcut your modernization journey. Established CoEs are available to help agencies with the planning and adoption of more efficient technology and a host of other solutions for IT modernization.

Make sure to avail yourself and your team of the wealth of training that's available to federal employees and contracting staff to get them up to speed including webinars and events. Continue to reiterate your mission and objectives for the transformation, teaching and training your team so that they become more comfortable and confident in the process.

Implement Programmatic Modernization

Flexibility will be fundamental to aligning your tools with continually shifting mission requirements while simultaneously paying down your technical debt. By approaching these challenges with a programmatic mindset and enterprise-level plan, you can solve for the 'big rocks' and avoid building up additional tech debt. It is crucial to focus some of your time on the longer-term, non-DevOps efforts to ensure that you are moving tooling in a smart, efficient way to keep up with the modernization cycle.

One programmatic approach that some agencies have found effective is to modernize and innovate in rapid, intense programmatic cycles on the enterprise-level (e.g., in six months cycles) alternating with modernization on their main mission priorities or larger initiatives. Other organizations have consolidated all of their tools and platforms to live within an automated pipeline ('Infrastructure-as-Code') so that when a platform is deployed, it also is delivered in a completely automated fashion, whenever possible.

Envisioning the Future State

Applying Artificial Intelligence (AI) and Machine Learning (ML)

When IT leaders look to the future, they should do so with an eye towards incorporating key capabilities and benefits of AI and ML into their ongoing modernization efforts such as using data gathered on the business side to provide insight into how they operate. As you operationalize moving forward with your pipelines, think about the tools that you possess to aggregate that data, then provide meaningful, useful information to your developers, operators, security, platform engineers, and site reliability engineers. Begin to apply levels of real-time AI and ML to data that can enable adjustments to how you do business within the organization.

Unlocking the Human Potential

People will continue to be the core of your transformation journey. Unlock the human potential of your operators, users, and other stakeholders by enabling them to come together so they can innovate faster and more collaboratively. Collaboration is currently a challenge for many of today's organizations because there are so many different silos for the way people work—in different systems and in different places. In order to solve tomorrow's problems, we will need to avoid technology for technology's sake and leverage technology that ensures people can work together to effectively solve big problems faster.

About GitLab

GitLab is a complete DevSecOps platform, delivered as a single application, fundamentally changing the way Development, Security, and Ops teams collaborate. GitLab helps teams accelerate software delivery from weeks to minutes, reduce development costs, and reduce the risk of application vulnerabilities while increasing developer productivity. GitLab provides unmatched visibility, radical new levels of efficiency and comprehensive governance to significantly compress the time between planning a change and monitoring its effect. Now, fast paced teams no longer have to integrate or synchronize multiple DevOps tools and are able to go faster by working seamlessly across the complete lifecycle.

GitLab delivers complete real-time visibility of all projects and relevant activities across the entire DevSecOps lifecycle. For the first time, teams can see everything that matters. Changes, status, cycle times, security and operational health are instantly available from a trusted single source of data. Information is shown where it matters most, e.g. production impact is shown together with the code changes that caused it. And developers see all relevant security and ops information for any change. With GitLab, there is never any need to wait on synchronizing your monitoring app to version control or copying information from tool to tool. GitLab frees teams to manage projects, not tools. These powerful capabilities eliminate guesswork, help teams drive accountability, and give everyone the data-driven confidence to act with new certainty. With GitLab, DevSecOps teams get better every day by having the visibility to see progress and operate with a deeper understanding of cycle times across projects and activities.

GitLab drives radically faster cycle times by helping DevSecOps teams achieve higher levels of efficiency across all stages of the lifecycle making it possible for Product, Development, QA, Security, and Operations teams to work at the same time, instead of waiting for handoffs. Teams can collaborate and review changes together before pushing to production. GitLab eliminates the need to manually configure and integrate multiple tools for each project. GitLab makes it easy for teams to get started, they can start with GitLab using one or two use cases they need to improve, and then begin their evolution to a single end-to-end experience for all of DevSecOps.

Only GitLab delivers DevSecOps teams powerful new governance capabilities embedded across the expanded lifecycle to automate security, code quality and vulnerability management. With GitLab, tighter governance and control never slow down DevOps speed.

GitLab leads the next advancement of DevSecOps. Built on Open Source, GitLab delivers new innovations and features on the same day of every month by leveraging contributions from a passionate, global community of 4800+ developers and millions of users. Over 100,000 of the world's most demanding organizations trust GitLab to deliver great software at new speeds.

Start your GitLab free trial
about.gitlab.com/free-trial

References

This whitepaper is partially based on the ATARC-hosted webcast "*DevOps: Powering Speed to Mission*", facilitated by Tom Suder, President and Founder of ATARC, and featuring Leo Garciga, Chief of JD-OI6 and Chief Technology Officer for the JIDO/DTRA, Rob Brown, Division Chief and Senior Solutions Architect with Infrastructure Enterprise (EID) at USCIS, and John Jeremiah, Subject Matter Expert at GitLab. https://about.gitlab.com/webcast/devops-speed-to-mission/

- NIST Cybersecurity Framework
  - https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
  - https://www.nist.gov/cyberframework/assessment-auditing-resources
- U.S. Federal Government Modernization Centers of Excellence Program Act
  - https://khanna.house.gov/sites/khanna.house.gov/files/Final%20Draft.pdf
- IT Modernization Centers of Excellence
  - https://coe.gsa.gov/
- Agile Aquisitions in Government
  - https://insights.sei.cmu.edu/devops/2015/05/devops-in-government-where-to-start.html
  - https://18f.gsa.gov/tags/agile/
  - https://18f.gsa.gov/2016/11/15/modular-procurement-state-local-government/
- GitLab DevSecOps Methodology Assessment
  - https://about.gitlab.com/resources/devsecops-methodology-assessment/

Related Reading

- *Speed to Mission* whitepaper
- *A Seismic Shift in Application Security* whitepaper


GitLab