



DevSecOps

Unifying teams for efficient, secure delivery.

In complex environments with distributed responsibilities, we build unified teams that share responsibility to consistently deliver and maintain secure applications and tools.

THE TCG APPROACH

TCG uses an iterative approach across four dimensions to ensure that an agency's DevSecOps practice is perfectly tailored to their environment.

Technologies: Identifying the nearest neighbor technologies to what's currently in use, so organizations don't need to upend their whole infrastructure.

Practices: Adopting and integrating practices in an order that will most positively impact the team at every step.

Zero-Trust: Planning and executing multiple iterations of Zero-Trust architecture based on the premises of elimination of implicit trust, continuous verification, and the assumption of a breach.

Team Building: Embracing collaboration, a core principle of DevSecOps, to work across the multiple vendors that perform development, security, and O&M.

WHY OUR APPROACH WORKS

Only pay for features that an organization needs:

A complete shift in an underlying platform, code repository, build process, and test process requires a large investment. Many times such a dramatic shift is unnecessary and wasteful. We help agencies assess their environment, find and implement the right tools, and, if a big change is needed, provide a solid foundation for making changes at the right pace.

Minimize risks: Use of neighboring technologies and iterative transition helps avoid errors that are difficult to diagnose and system outages that often accompany new tools and processes.

Maintain institutional knowledge: There is valuable institutional knowledge at agencies in the current architecture and maintenance. Reconstructing this knowledge for a new infrastructure is costly and time consuming.

CONTINUOUS DELIVERY PIPELINE

CONTINUOUS DEVELOPMENT

Release products quickly in small increments for continual improvement after deployment.

CONTINUOUS INTEGRATION

Automate integration of code changes to eliminate bugs and establish sustainable development pace.

CONTINUOUS TESTING

Develop tests before writing the code or feature to build in quality from the beginning.

CONTINUOUS DELIVERY

Deploy integrated changes with the press of a button to save time and ensure consistent delivery of reliable products to end users.

SECURITY

Scan, test, and review code for security issues from the beginning of the development cycle.

