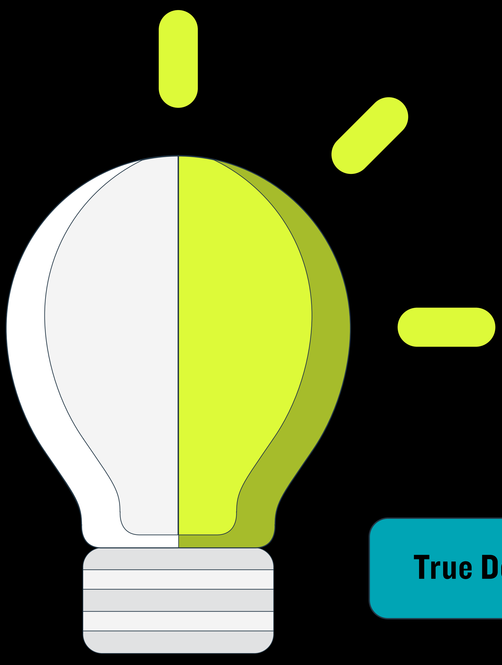
Booz Allen®



Pretend DevSecOps

True DevSecOps

CONTINUOUS INTEGRATION (CI)

Pretend DevSecOps

+ Daily builds done — but often broken

True DevSecOps

Confidence in infrastructure 24/7

+ Daily builds include infrastructure
+ Teams ensure the build is sound

CONTINUOUS DEPLOYMENT

Pretend DevSecOps

+ Relies on text-based deployment instructions
+ Process is slow and open to errors

True DevSecOps

Almost 100% uptime, lower cost

+ Automated daily deployments incorporate every enhancement
+ For necessary push-button deployments, container management software (Kubernetes, Rancher AgroCD) ensures fast, orderly Process

INFRASTRUCTURE AS CODE (IAC)

Pretend DevSecOps

+ Environments configured via the console
+ Disparate steps and guides leave test and ops environment out of sync

True DevSecOps

Environment parity ensures efficiency

+ Evironments built out with one click using IaC tools (Ansible, Chef, Terraform)
+ Infastructure deployments are automated (GitOps)
+ Standardization saves time, lower errors

SECURITY

Pretend DevSecOps

+ Run scans before deployment to address high-priority findings only
+ Deployed code therefore contains faults to be fixed “when there’s time”

True DevSecOps

Less risk, faster delivery at less cost

+ Automated security tools scan code, finding and correcting issues before deployment
+Deployment systems are scanned daily (IaC process)
+ Daily build includes continuous monitoring

HOTFIX

Pretend DevSecOps

+ Bugs accidentally deployed need manual resolution
+ Task goes on the to-do list or takes time from that day’s development schedule

True DevSecOps

Fast fixes speed progress, lower risk

+ Code is strictly version-controlled
+ Issues are rapidly identified down to “commit” stage
+ Capability can be instantly rolled back to previous healthy state for double protection

DEPLOYMENT DOWNTIME

Pretend DevSecOps

+ System goes down for routine developments
+ Hours of downtime creates frustration and delays

True DevSecOps

Safer deployments, little or no downtime

+ Advanced release strategies (traffic splitting, blue-green and canary deployments) accelerates process

AUTOMATION

Pretend DevSecOps

+ Pipeline tool chain is automated only for deployment
+ Developers need to do manual integrations for each application or service
+ Separate processes slow development

True DevSecOps

Single-click processes accelerate delivery

+ Enhanced tool chain (scripts, plug-ins, glue code) ensures fully automated, end-to-end integration
+ Single-click build and deployment

OPS ACCREDITATION

Pretend DevSecOps

+ Capability releases go to operations quickly, but deployment is held up for security approval
+ Manual approval takes time and is open to error

True DevSecOps

On-demand ops save time, lowers risk

+ Security compliance and checks are integrated in automated pipelines for rapid capability deployments
+ Ensures compliance, eliminates manual secruity approvals