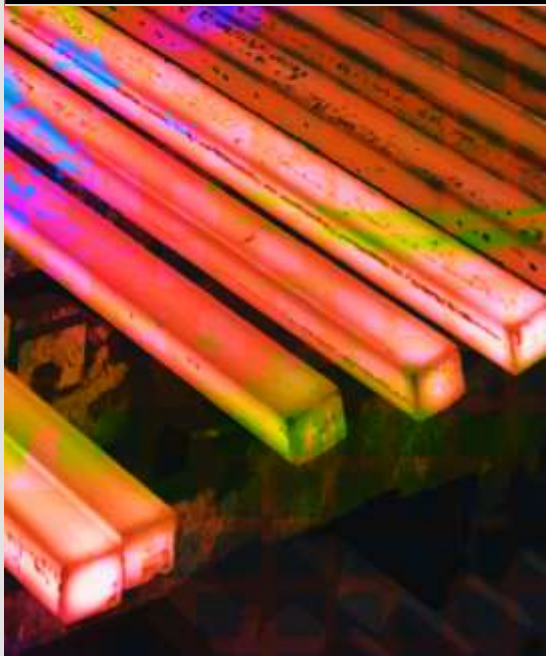
Carnegie Mellon University
Software Engineering Institute

Top 5 Challenges to Overcome on Your DevSecOps Journey



MAY 02, 2023

Hasan Yasar
Technical Director and Faculty Member

Joseph Yankel
DevSecOps Initiative Lead and Faculty Member

Agenda

- Overview of DevSecOps
- Top 5 DevSecOps Challenges and Actions
- Sustaining DevSecOps Platform
- What is Next?

Presentation Name

Overview of DevSecOps

Carnegie Mellon University
Software Engineering Institute

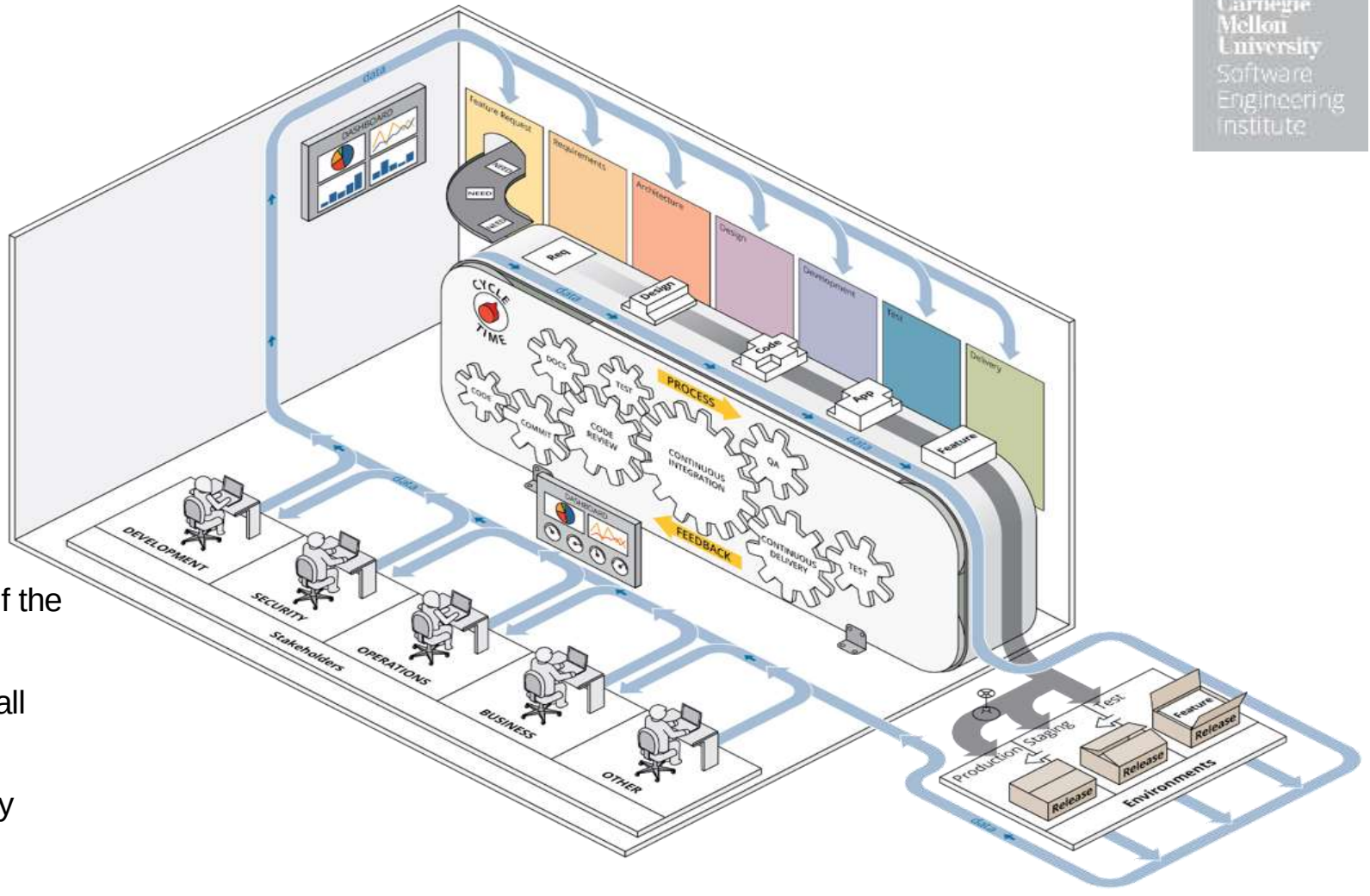
DevOps is a set of principles and practices emphasizing collaboration and communication between software development teams and IT operations staff along with acquirers, suppliers, and other stakeholders in the lifecycle of a software system.

DevSecOps builds upon DevOps principles with additional focus on security activities throughout the lifecycle phases of requirements, design, coding, testing, delivery, deployment and incident response.

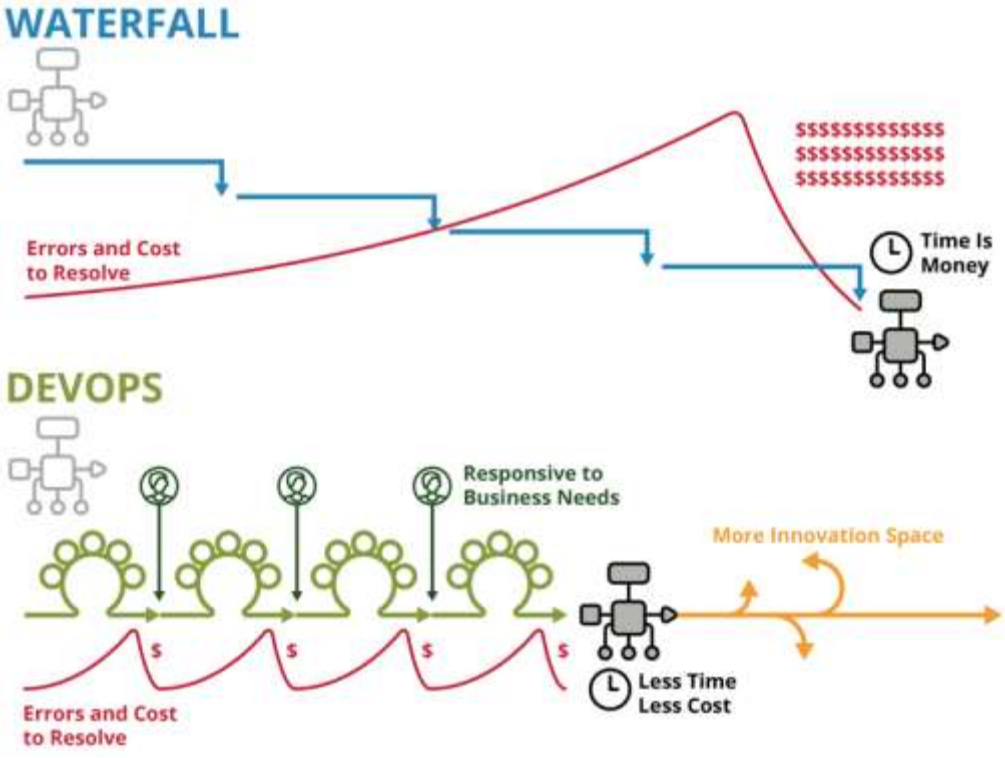
Mature DevOps practices are constantly testing, deploying and validating that software meets every requirement and allows for fast recovery in the event of a problem. As a result we can easily say,

"DevSecOps is DevOps done right"

- Feature to deployment
- Iterative and incremental development
- Continuous feedback
- Metrics and measurement
- Automation in every phase of the SDLC
- Complete engagement with all stakeholders
- Transparency and traceability across the lifecycle



Key benefits of DevSecOps



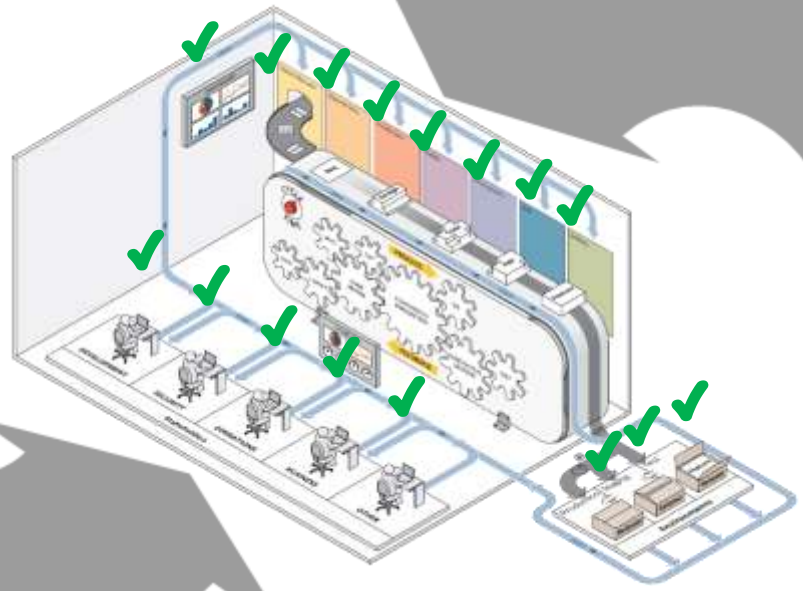
- Reduced security errors during deployment
- Reduced time to deploy and response incident on time
- Repeatable/automated steps
- Continuous availability of pipeline and application
- Increased time for learning new concepts
- Responsiveness to business needs
- Increased stability and quality

Think **Security** from Inception to Deployment, and improve every delivery by gathering *all metrics.*

Carnegie Mellon University Software Engineering Institute

Data…

- Attack Vector Details (e.g., IP, Stack Trace, Time, Rate of Attack)
- Server Disk Space, Load and Process Monitoring
- Application Performance
- Maximize Monitoring
- Change in Size to Code Base
- Most Active Code Contributors
- Most Changed Code Areas



Data…

- Deployment Frequency
- Change Lead Time and Volume
- Change Failure Rate
- Mean Time To Recovery (MTTR)
- Mean Time to Detection (MTTD)
- Issue Volume and Resolution Time
- Time to Approval
- Time to Patch Vulnerabilities
- Development and Application Logging Availability
- Retention Control Compliance
- SAR Findings

DevSecOps Challenges

DevSecOps Challenges

1. Lack of security assurance at the business and project levels
2. Organizational barriers related to collaboration, tooling, and culture
3. Impact to quality because security is not a priority while systems are getting more complex
4. Lack of security skills for developers, business stakeholders, and auditors
5. Insufficient security guidance due to lack of resources, standards, and data

CHALLENGE #1: Lack of Assurance

- Industry lacks assurance models
- Business lacks assurance of security
- Project lacks assurance of security

ACTION:

- Don’t wait for an industry standard to emerge
- Join informal working groups with industry peers
- Attend conferences and network
- Share your experience with lesson learned
- Work with others to extend the body of knowledge and establish best practices

CHALLENGE #1: Lack of Assurance

Carnegie Mellon University Software Engineering Institute

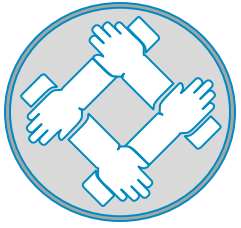
- Industry lacks assurance models
- **Business lacks assurance of security**
- Project lacks assurance of security

ACTION:

- Focus on fundamentals: What are the threats? What are the business drivers? Balance the two.
- Align with business needs (time to market, cost savings, resilience)
- Conduct external audits
- Understand the business context
- Identify, link, and rank business and technical risks
- Get security requirements in early
- Define the risk mitigation strategy
- Educate top management and get them on board
- Engage more senior technical people first to work with security teams
- Make security part of senior technical reviews; organically spread the word

CHALLENGE #1: Lack of Assurance

- Industry lacks assurance models
- Business lacks assurance of security
- Project lacks assurance of security



ACTION:

- Map reporting data from tools to form a continuous view of value
- Run security tools on all code to measure code quality and standards
- Review code changes for security and document approval prior to release
- Use dedicated testing resources in the case of significant changes
- Track all changes and approvals for incident purposes
- Conduct code reviews
- Expose security team to your metrics and data

CHALLENGE #2: Organizational Barriers

Carnegie Mellon University Software Engineering Institute

- Poor stakeholder collaboration
- Integrating pipeline security
- Making security a priority

ACTION:

- Document your current state; you are going to have some type of silos around Development, Infrastructure, and Security
- Start building collaboration between Security and Dev and Ops teams
- Be prepared: people generally don't want to change their culture/workflow
- Make sure everyone gets on the same page around the importance of security (from Execs to DevSecOps teams)
- Instill a continuous security mindset
- Focus on partnership not unhealthy conflict; destroy the blame culture
- Get stakeholders to agree on a shared a vision for a project
- Balance workload among teams involved
- Put security people into Dev teams

CHALLENGE #2: Organizational Barriers

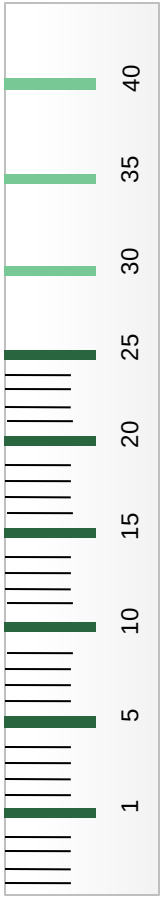
Poor stakeholder collaboration

Integrating pipeline security

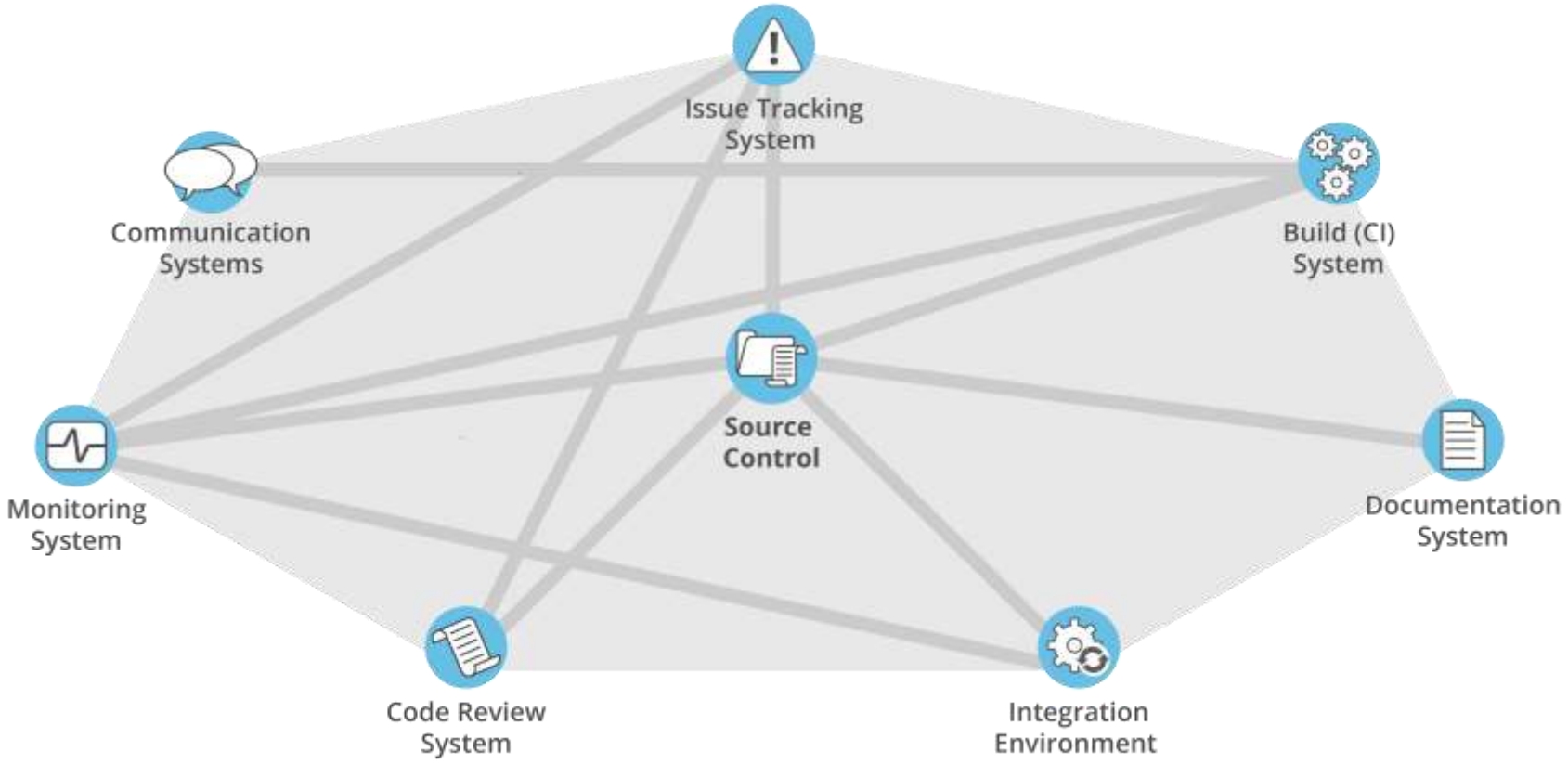
Making security a priority

ACTION:

- Integrate your process with TM, SAST, DAST, IAST
- install security requirements traceability
- Metrics can help
  - MTTR, MTTD, vulnerability escape rate, repeated incident root cause, time to deploy the app from Dev to Prod
- Look at different approaches
  - Abuse cases, architectural risk analysis, application penetration testing
- Design for security
  - Fail securely and fail safe defaults
  - Least privilege
  - Defense in depth
- Automate where possible
  - IaC, virtualization, containers, and load balancing
  - Configuration management
  - Continuous application and performance monitoring



Integrating pipeline security



CHALLENGE #2: Organizational Barriers

Poor stakeholder collaboration

Integrating pipeline security

Making security a priority

ACTION:

- Use evangelists to drive culture change
- Explain why security is an important, shared responsibility, and its impact
- Embed security into ops escalation
- Invite security to post-mortems
- Create a plan in small parts; start with a pilot and be mindful of cross-team resource constraints
- Keep it simple; don't overwhelm the system. If there are too many things to do, the chances this plan will fail.
- Incrementally chase real risk and threats first
- Test whether your organization is ready for the culture change; no single technology/tool will get you DevSecOps

CHALLENGE #3: Lack of Quality

Security brought in too late

Lack of confidence in release

System complexity

ACTION:

- Start getting security and compliance requirements in early
- Tie compliance objectives into providing assurance back to the business
- Test compliance against security policies to identify gaps
- Define a risk mitigation strategy early

CHALLENGE #3: Lack of Quality

- Security brought in too late
- Lack of confidence in release
- System complexity

ACTION:

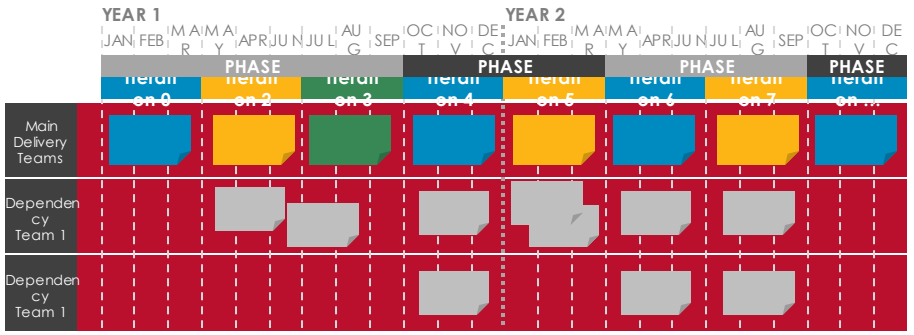
- Instill risk based security testing
- Move the conversation from CABs and phase gates to compliance driven releases
- Automate reporting for compliance violations and stop the pipeline when threshold is exceeded or policy not met
- Move toward frequent, automated audits
- Audit yourself to demonstrate compliance with policies or regulations
- Establish security requirements traceability

CHALLENGE #3: Lack of Quality

- Security brought in too late
- Lack of confidence in release
- System complexity

ACTION:

- Identify proxy metrics for complexity
  - Number of issues in production
  - Time to deploy application
- Drive security policies into production by integrating security tasks in early stages of DevSecOps pipeline

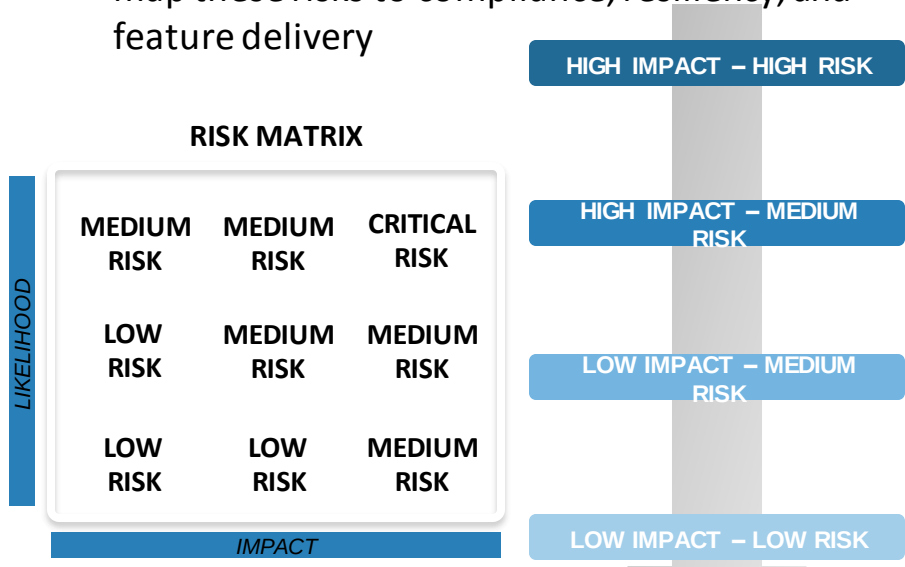


CHALLENGE #4: Lack of Security Skills

- Business lacks security skills
- Developers lack security skills
- Auditors lack security skills

ACTION:

- Shift the conversation to risk and quality
- Service and protect the business interests to lower risk
  - Identify risk/security value
- Identify architectural risk and uncertainty; map these risks to compliance, resiliency, and feature delivery



CHALLENGE #4: Lack of Security Skills

Business lacks security skills

Developers lack security skills

Auditors lack security skills

ACTION:

- Keep the end game in mind; build a collaborative security culture
- Implement compliance automation as a way to drive business thinking into the SDLC
- It's not a checkbox; Security training once a year has limited effectiveness
- Aim for attitude and behavior; simply providing better technical training alone won't change attitudes
- Motivate and unblock the path toward the goal (remove task ambiguity, set clear role targets, don't overload)
- Aim for long term retention; apply learning in context repeatedly
- Rotate experts on the team where possible

CHALLENGE #4: Lack of Security Skills

- Business lacks security skills
- Developers lack security skills
- Auditors lack security skills

ACTION:

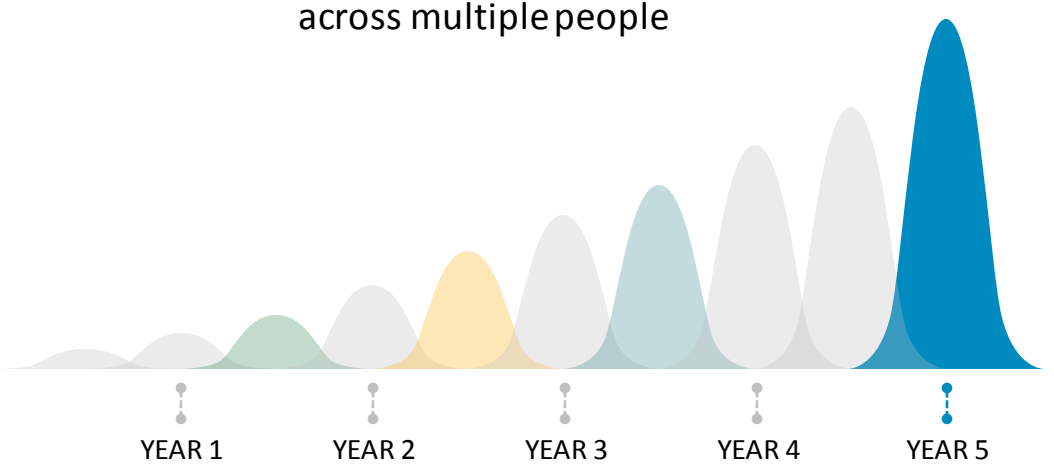
- Build working relationships and collaboration across silos
- Make security as part of informal discussions
- Provide cross functional training for both technical and compliance domains
- Integrate low disruption workflows
- Get familiar with some common standards and frameworks (OWASP Top 10, NIST 800-53, ISO 27001

CHALLENGE #5:Insufficient Security Guidance

- Lack of security resources
- Lack of security standards
- Lack of proactive monitoring

ACTION:

- Aim for long term sustainability; when you come back a couple of years after deployment is the change still there?
- Start by introducing a policy and assess your gaps; grow from there
- Map policies to domain specific procedures (development, testing)
- The goal is to spread security responsibility across multiple people



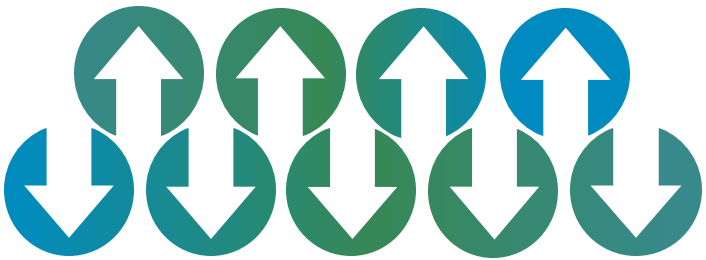
CHALLENGE #5:Insufficient Security Guidance

Carnegie Mellon University Software Engineering Institute

- Lack of security resources
- Lack of security standards
- Lack of proactive monitoring

ACTION:

- Don’t go big bang
- Start with a well known framework like OWASP Top 10; create a few policies around that
- Aim low hanging fruit (CI or testing, for example); measure security against your initial policies
- Grow from there by looking upstream and downstream for the next easiest implementation
- Bake policies into the workflow to avoid regression



CHALLENGE #5:Insufficient Security Guidance

- Lack of security resources
- Lack of security standards
- Lack of proactive monitoring

ACTION:

- Start with Ops log monitoring before attempting expensive tools
- Create feedback loop from Ops back to Development
- Update security documentation including trust boundaries, new threats, and component verification.

Sustaining your DevSecOps environment

Carnegie Mellon University Software Engineering Institute

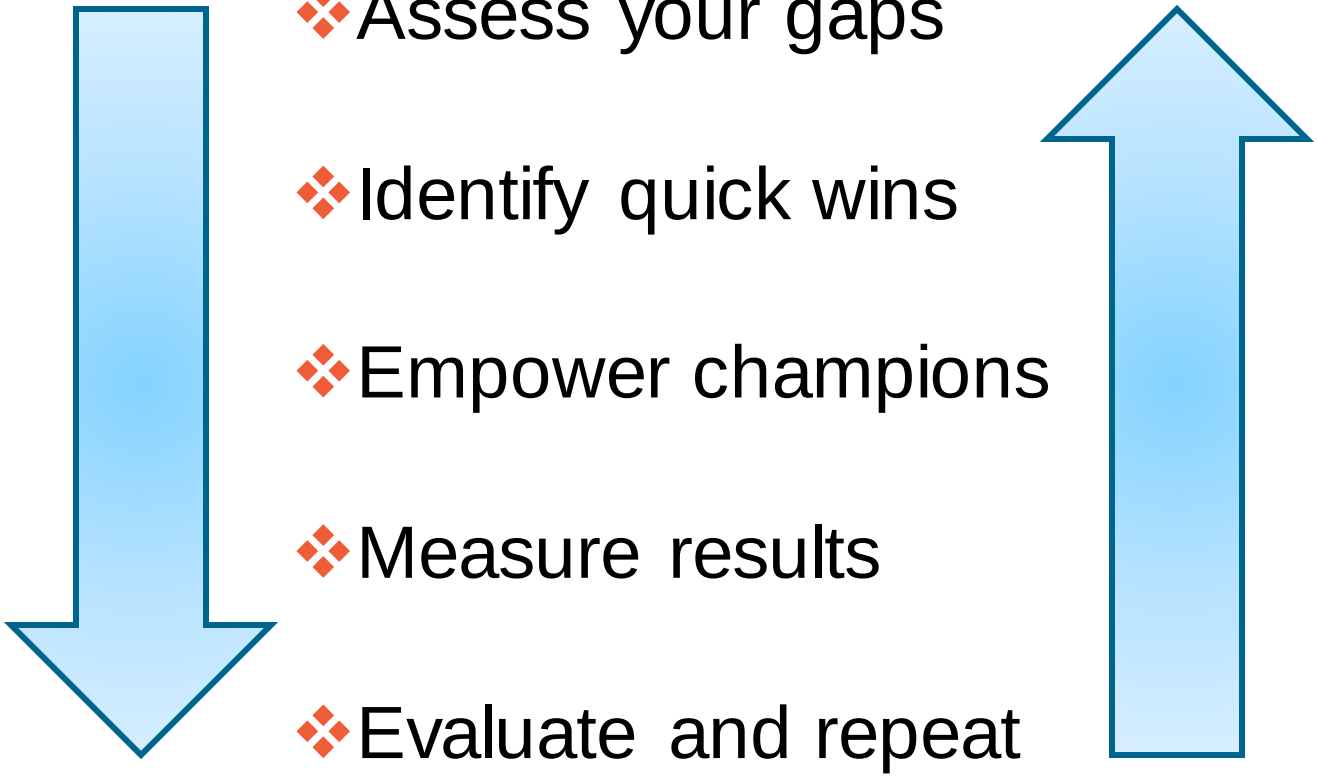
Effective Usage

- Train Users and build DevOps skills
- All stakeholders access
- Playbook/Developers guidance
- Project startup guidance
- Project Architectural Guidance
  - Common Services,
  - Common Security approach
  - Architectural patterns
  - Test methods
- DevOps environment usage policy
  - Build and Deployment Strategies

Maintaining (cost/update)

- Updating the environment (new version or security patches)
- Supporting new tools
- Adding/setting up new projects
- Operational Support
  - Base Image, OSS Support, Test harness, Temp Environment Creation
- Pipeline orchestration
- Securing pipeline
- Usage meter/billing support
- Auditability/log and data collection

What Is Next?



- Assess your gaps
- Identify quick wins
- Empower champions
- Measure results
- Evaluate and repeat

Carnegie Mellon University Software Engineering Institute

For More Information

DevSecOps: https://www.sei.cmu.edu/go/devops
DevOps Blog: https://insights.sei.cmu.edu/devops
Webinar Series: https://www.sei.cmu.edu/publications/webinars/
Podcast Series: https://www.sei.cmu.edu/publications/podcasts/

Contact Information

Hasan Yasar
Technical Director, Adjunct Faculty Member
Continuous Deployment of Capability,
Software Engineering Institute | Carnegie Mellon University
hyasar@cmu.edu

Joseph Yankel
Initiative Lead, DevSecOps Innovations
Continuous Deployment of Capability,
Software Engineering Institute | Carnegie Mellon University
jdyankel@sei.cmu.edu