IT/Dev CONNECTIONS

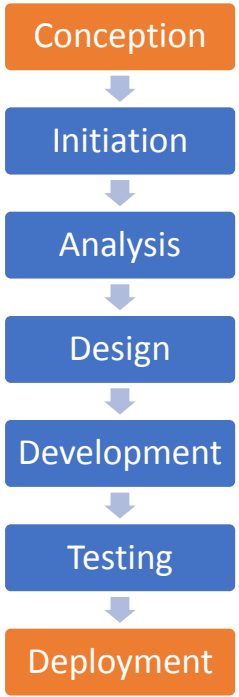
How to Get Started with DevSecOps

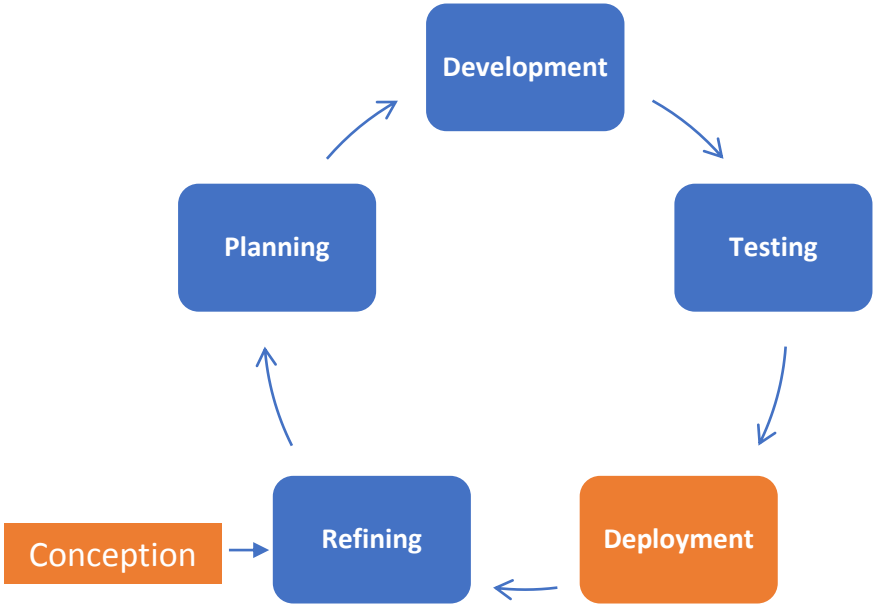
Andrei Bezdedeanu
VP of Engineering
CYBRIC

Delivered by KNect365 TMT an Informa business

IT/Dev CONNECTIONS

Waterfall vs. Agile



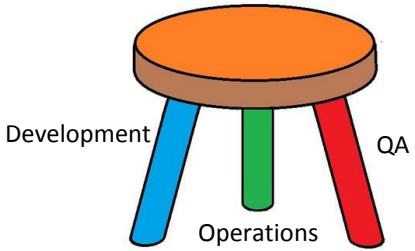


Delivered by KNect365 TMT an Informa business

Agile Manifesto

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

DevOps



DevOps is the combination of **cultural philosophies**, **practices**, and **tools** that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.

IT/Dev CONNECTIONS

Challenges and Enablers to DevOps

Challenges

- Organizational Structure
- Lack of understanding
- Goals misalignment
- Personal egos
- Technical incompetency
- Corporate politics

Enablers

- Executive support
- Leading by example
- Champions
- Automation, automation, automation
- Technical competency
- Early success

IT/Dev CONNECTIONS

Components of a Successful DevOps Strategy

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration
- Shared Accountability

Benefits of DevOps

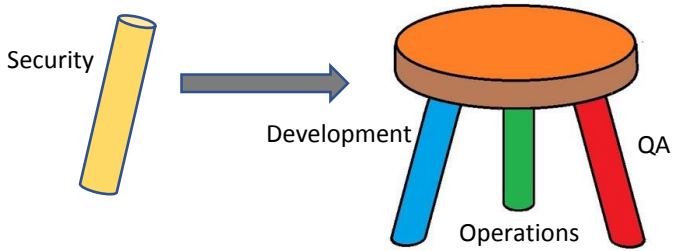
- Collaboration - alignment between development and operations teams; handoff friction is reduced, common goals and objectives
- Fluid responsiveness - real-time feedback and greater efficiency; changes and improvements can be implemented quicker
- Shorter cycle time - efficiency and communication between teams shortens cycle time; new code can be released more rapidly
- Better Quality – bugs are discovered and fixed more quickly, goal of every Agile sprint is to deliver working, quality software

DevOps vs SecOps

- **DevOps does not mean more secure applications!**
- Growing chasm between DevOps and SecOps
  - Security testing too late in the application delivery cycle
  - Testing done by SecOps team not familiar with the code base
  - Security not often part of initial design
  - No threat modeling
  - Fixing vulnerabilities in conflict with sprint goals
- Increased frustration for both teams
- Finger pointing

DevSecOps

- A bridge between fast and secure software development
- A cross-functional team composed of Development, Operations, Security and QA
- United security and engineering culture
- Security at the speed of DevOps, leveraging automation to fullest extent
- Making Security an integral part of application design



Keys to a Successful DevSecOps Transformation

- Insert security into the DevOps culture
- Break down the “walls”
- Design with security in mind
- Embed security throughout SDLC
- Empower developers with appropriate security tools
- Treat everything as code
- Manage Security Posture after deployment
- Include Development and Operations in Incident Response

IT/Dev CONNECTIONS

Challenges and Enablers to DevSecOps

Challenges

- Organizational structure
- Developers lack knowledge of security tools & practices
- Security engineers lack understanding of development

Enablers

- Executive support
- CI/CD
- Infrastructure as code
- Education

“Shift Left”

- Empower development team with tools, process and best practices
- Security checks in the IDE
- Scan code on commit
- Scan output of build process
- Scan container images built
- Scan applications deployed to non-production environments
- Scan applications in production
- Scan infrastructure and network
- Check cloud and network configurations

Automation and Orchestration

- A proper DevSecOps strategy has little room for manual testing and scanning
- Functional testing and Penetration testing can still happen, but…
- Automate all scans that can be automated
- Orchestrate the process by integrating into code repositories, build and deployment pipelines

IT/Dev CONNECTIONS

What Scanning Tools Do We Use?

SAST

Analyze application source code, byte code and binaries for coding and design conditions that are indicative of security vulnerabilities.

SCA

Identify risks from open source libraries and commonly used frameworks, covering both known security vulnerabilities and license risk.

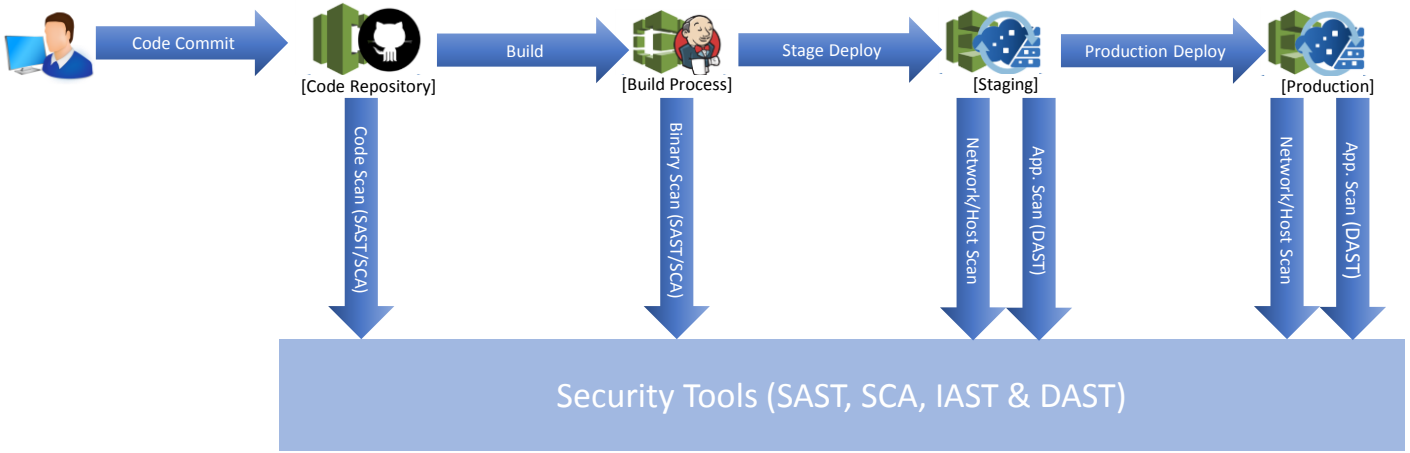
DAST

Detect conditions indicative of a security vulnerability in an application in its running state (outside-in)

IAST

Analyze application behavior in the testing phase by instrumenting the runtime engine to get insight into the application's logic flow, data flow and configuration. Monitor test attacks initiated by a DAST attack inducer, and then report on the attacks that resulted (or might result) in an application's exploit.

From Code Commit to Deployment



Containerization & Cloud Transformation

- Integrating security into continuous delivery can be challenging
- Developers and IT ops are rarely trained in security
- Security teams don't necessarily know how to code or administer servers
- For DevSecOps to work, everyone involved in continuous delivery needs to speak the same language and work with the same environment

- Infrastructure as code provides a common framework
- Containers are the key to achieving automated, predictable security operations
- Container security is a fast maturing space (NIST 800-190)

Data Aggregation & Single Pane of Glass

Security scanning tools provide rich set of vulnerability data

- **Normalize** data into a single format
- **Consolidate** into single issues of the same type
- **Deduplicate** across tools with overlapping coverage
- Allow users to **override** based on what they know
- **Correlate** Static and Dynamic issues
- **Prioritize** exploitable vulnerabilities
- **Minimize** exposure and risk
- **Remediate** quickly

How fast, How bad, How expensive?

The three questions most asked:

1. How fast can we find out if/when we have a problem?
2. How bad is it (what is the risk)?
3. What is it going to take to fix it?

IT/Dev CONNECTIONS

Detection & Remediation Metrics

Internal Rate of Detection (IRD)

- The time it takes to find vulnerabilities
- Usually measured as the time between last scan and the first scan that identified a new issue
- Measures the risk of having a vulnerability and not knowing about it
- More frequent scanning = lower IRD

Internal Rate of Remediation (IRR)

- The time it takes to fix vulnerabilities
- Usually measured as time between detection and remediation
- Measures effectiveness of fixing vulnerabilities once found
- Addressing issues immediately = lower IRR

IT/Dev CONNECTIONS

Defect Metrics

Severity

- Degree of impact a vulnerability has on the development or operation of a component or application being tested
- The Common Vulnerability Scoring System (CVSS) is a free and open industry standard for assessing the severity of security vulnerabilities
- CVSS indicates ease and impact of exploit

Confidence Level

- Level of confidence in the existence of the vulnerability and also the credibility of the technical details of the vulnerability

Defect Density

- Number of defects (vulnerabilities, bugs) per thousands of lines of code

Cost of Remediation

Multi-variable calculation without a common standard/definition

Variables that can influence Cost of Remediation:

- IRD & IRR
- Severity
- Confidence Level
- Correlation & Exploitability

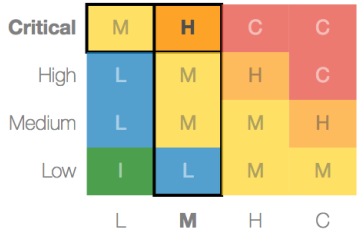
IT/Dev CONNECTIONS

Application & Enterprise Risk

Various risk estimation frameworks available…

OWASP Risk Rating Methodology

Risk = **Likelihood** * **Impact**



Likelihood Factors

- Threat Agent Factors (skill, motive, opportunity, size)
- Vulnerability Factors (ease of discovery, ease of exploit, awareness, intrusion detection)

Impact Factors

- Business (financial damage, reputational damage, non-compliance, privacy violation)
- Technical (loss of confidentiality, integrity, availability, accountability)

Delivered by KNect365 TMT an Informa business

IT/Dev CONNECTIONS

Incident Response & Management

DevSecOps is a good idea…
Early vulnerability detection and proper hygiene are good ideas…

But breaches will happen!!!
Efficient Incident Response is key when that occurs…

Agile and DevSecOps to the rescue

- DevSecOps provides a shared accountability culture
- Use information collected prior to the breach in development, testing and production
- Use available correlations to efficiently find problems in code and fix them
- Use existing automation framework to provide attestation of any fix
- Plan remediations and patching immediately or plan as part of next sprint

IT/Dev CONNECTIONS

The journey starts where you are today

There are no prerequisites for DevSecOps

- You can start with code security (start left)
- You can start with dynamic application scanning (start right)
- You can start ensuring that open source component risks are minimal
- You can start by just scanning container images for known vulnerabilities, or…
- You can put in place an end-to-end comprehensive strategy

Either way, DevSecOps can enable and accelerate the journey

IT/Dev CONNECTIONS

No Excuses!!!

We don't have CI/CD

We don't have any vulnerabilities

We are not ready

We don't have any scanning tools

We don't have resources

We don't have time

Security is important, but not a priority

We are not mature enough

We are not DevOps

Delivered by KNect365 TMT an Informa business

IT/Dev CONNECTIONS

Q & A

Email: andrei@cybric.io
LinkedIn: @andreibezdedeanu
Twitter: @abezdedeanu