



VIRGINIA DEPARTMENT OF  
SOCIAL SERVICES

# DevSecOps

## An Implementation Strategy With a Focus on Cultural Implications

6<sup>th</sup> Annual COV Information Security Conference  
Richmond, Virginia  
April 12, 2019

# Presenters



**Eddie McAndrew**  
COO  
AIS Network  
(804) 239-5185

Email:  
[eddie.mcandrew@aisn.net](mailto:eddie.mcandrew@aisn.net)



**Barry Davis**  
CISSO  
Virginia Dept. of Social Services  
(804) 726-7153

Email:  
[barry.davis@dss.virginia.gov](mailto:barry.davis@dss.virginia.gov)


# Agenda

---

- Introduction
- DevOps
- DevSecOps & Process
- DevSecOps Tools
- Summary
- Q&A

# DevOps

A person is seen from the side, wearing a yellow sweater, working at a desk. They are using a laptop in the foreground and several large monitors in the background. The laptop screen shows a complex interface with charts and data. The background monitors display code or system logs. The desk is cluttered with various items, including a white box, a pair of glasses, and a small bowl of dark berries. The overall scene is dimly lit, with the primary light source being the screens.



"DevOps is a set of software development practices that combines software development and information technology operations to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives."  
([Wikipedia](#))

# What Is DevOps?

# What Is DevOps?



Tools and practices employed to drive high velocity deployment of applications



Key component of value proposition behind going to the cloud



Drives Continuous Integration/Continuous Deployment (CI/CD)



Intended to drive innovation/continuous learning, high-quality applications through flexibility and enhanced competitiveness

# Key Elements

## Infrastructure as Code

Defining and managing system configuration through code that can be versioned and tested in advance, to increase the speed of building systems and offering efficiencies at scale.

## Continuous Delivery

Using Continuous Integration and test automation to build pipelines from development to test and then to production.

## Continuous Monitoring and Measurement

Creating feedback loops from production back to engineering, collecting metrics and making them visible to everyone to understand how the system is actually used, and using this data to learn and improve.

# DevOps CI/CD Driving Innovation

---

Integrating at  
the end of the  
life cycle is no  
longer sufficient!

## Continuous Integration – Continuous Delivery





# Continuous Integration Continuous Delivery

---

## Continuous Integration Continuous Delivery

---

Relies on the automation of routine work

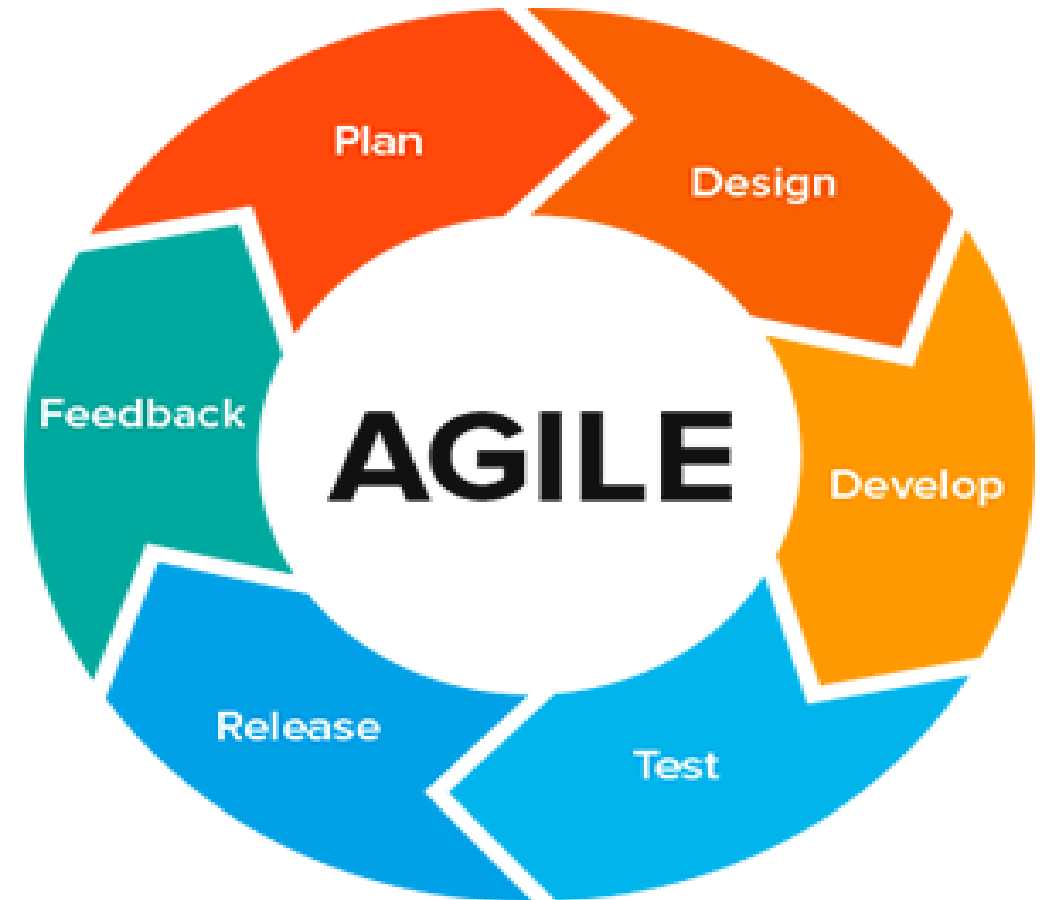
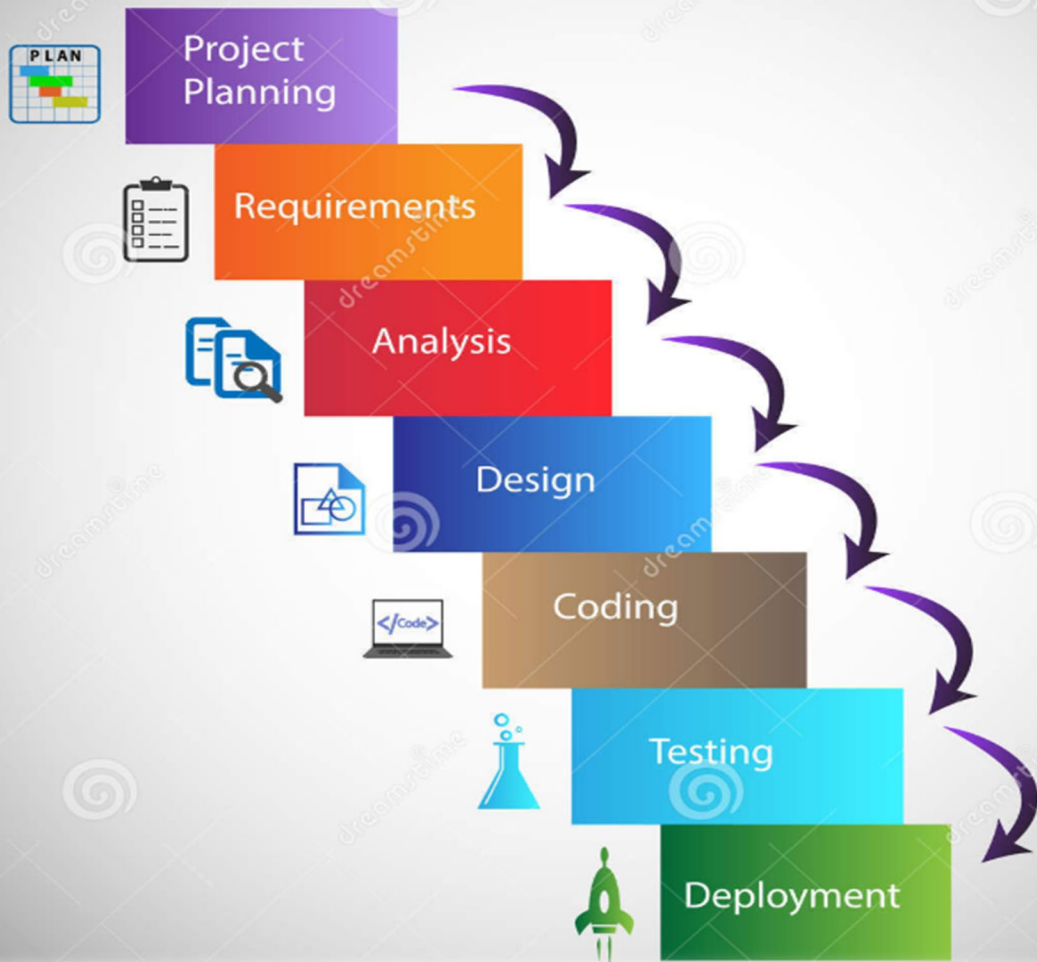
Transforming code changes

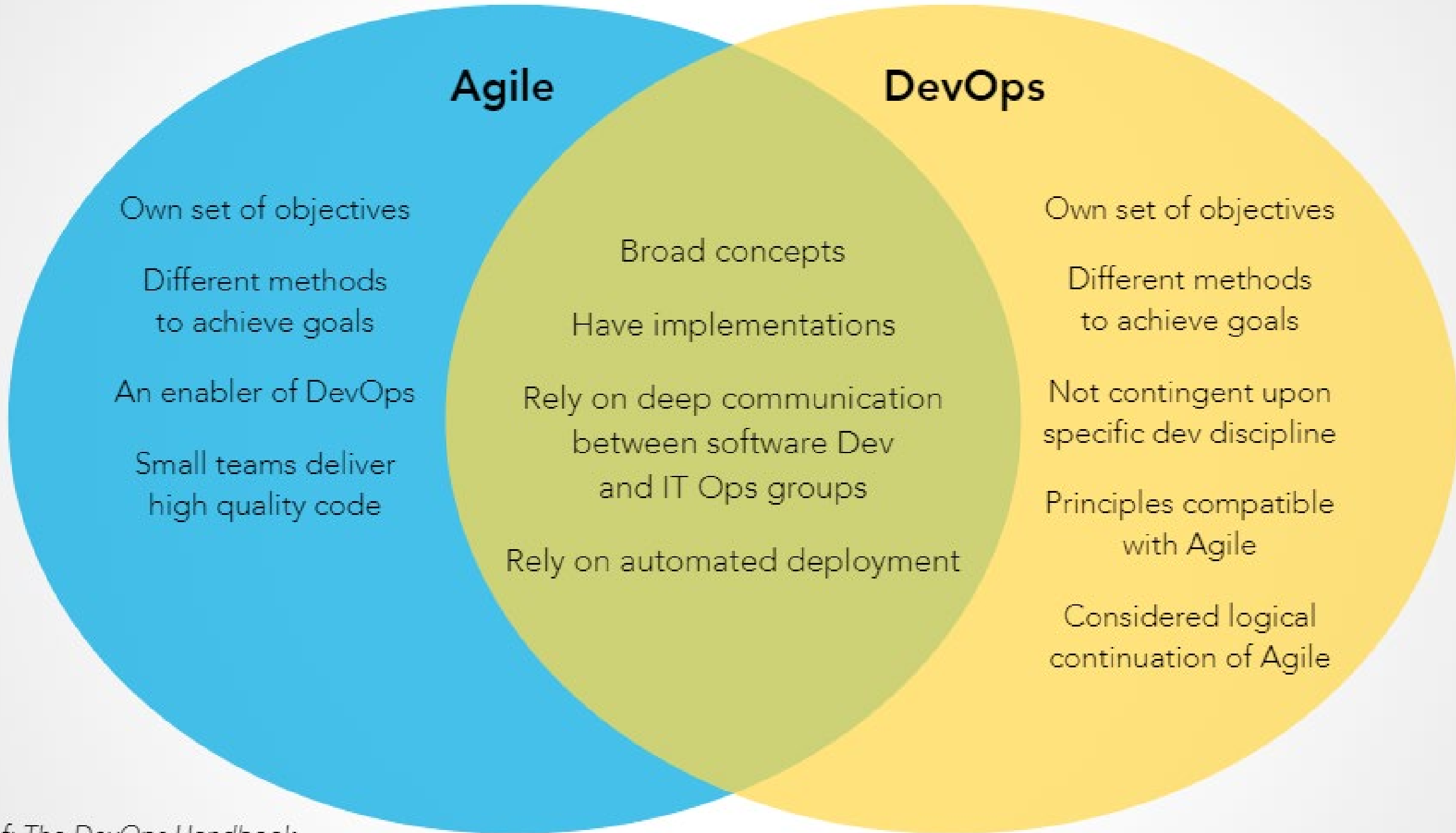
Delivering tested code into production

- ▶ Is a collection of technologies and practices
- ▶ Releases new code changes while keeping things stable

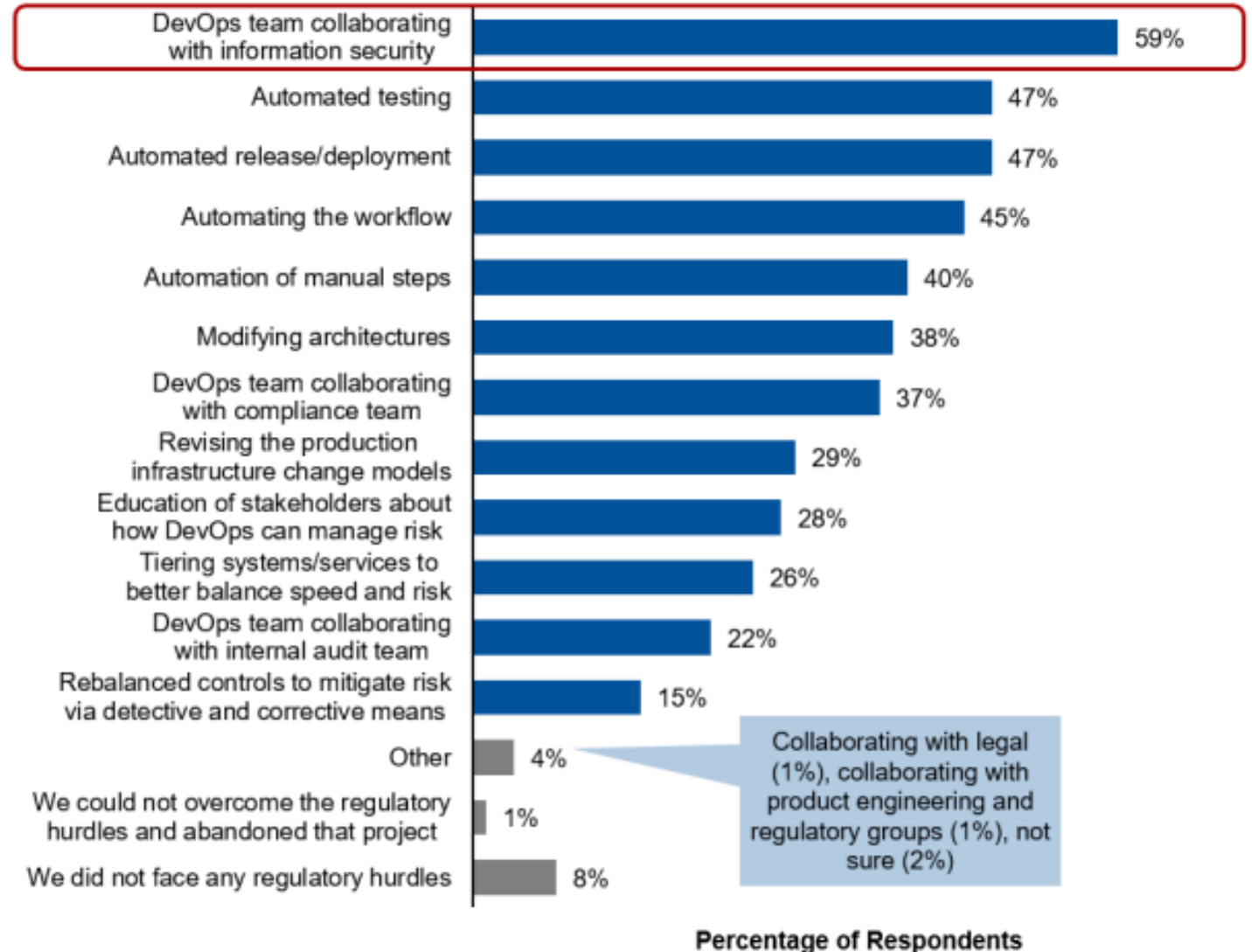
# Comparing Development Models

## Waterfall-Model





# Hurdles to Using DevOps in Regulated Situations



Base: n = 78 Gartner Research Circle Members who use DevOps approach and comply with regulations and/or obligations Q05. Did your organization employ any of these strategies to overcome these and/or other hurdles specific to using DevOps in regulated situations?

© 2017 Gartner, Inc.

A woman with glasses is looking at a laptop screen in a dimly lit office. The background is blurred, showing other people working. The text "DevSecOps & Process" is overlaid on the image in a white, outlined font.

# DevSecOps & Process

# DevSecOps: The Why and the What



“regardless of the software development and lifecycle management approach, security needs to be built into the software, not bolted on after the fact”

Faster deployment, rapid and continuous updates and rollout lead to what?

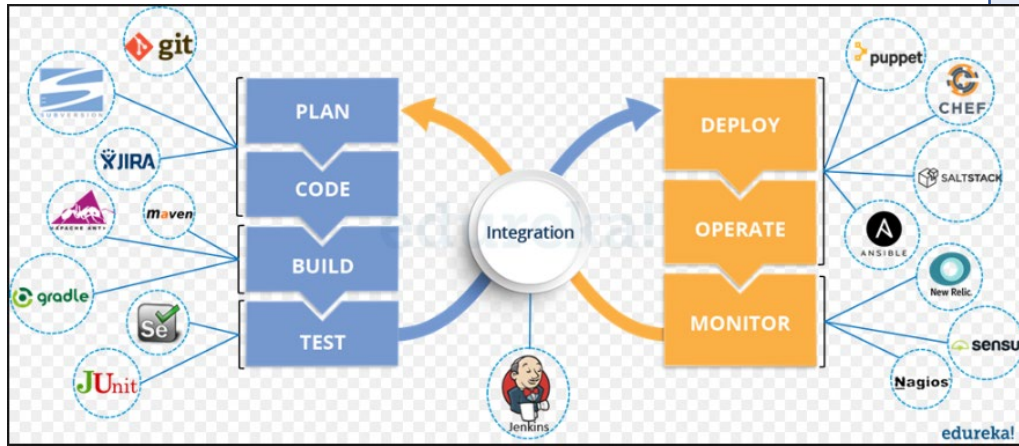
- More potential vulnerabilities
- Greater potential risk
  - So, to drive speed, flexibility & innovation securely -> DevSecOps

DevSecOps – Bridging Agility & Security

DevSecOps consists of the tools, frameworks and principles for adapting to a high velocity environment

- Driving *enabled* innovation, flexibility and competitiveness *securely*...

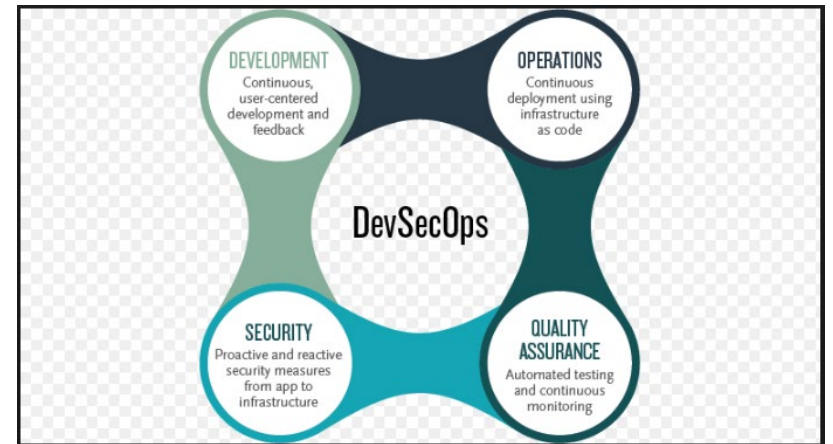
# Key Elements of DevSecOps



Culture

Process

Technologies



# Traditional Security v. DevSecOps

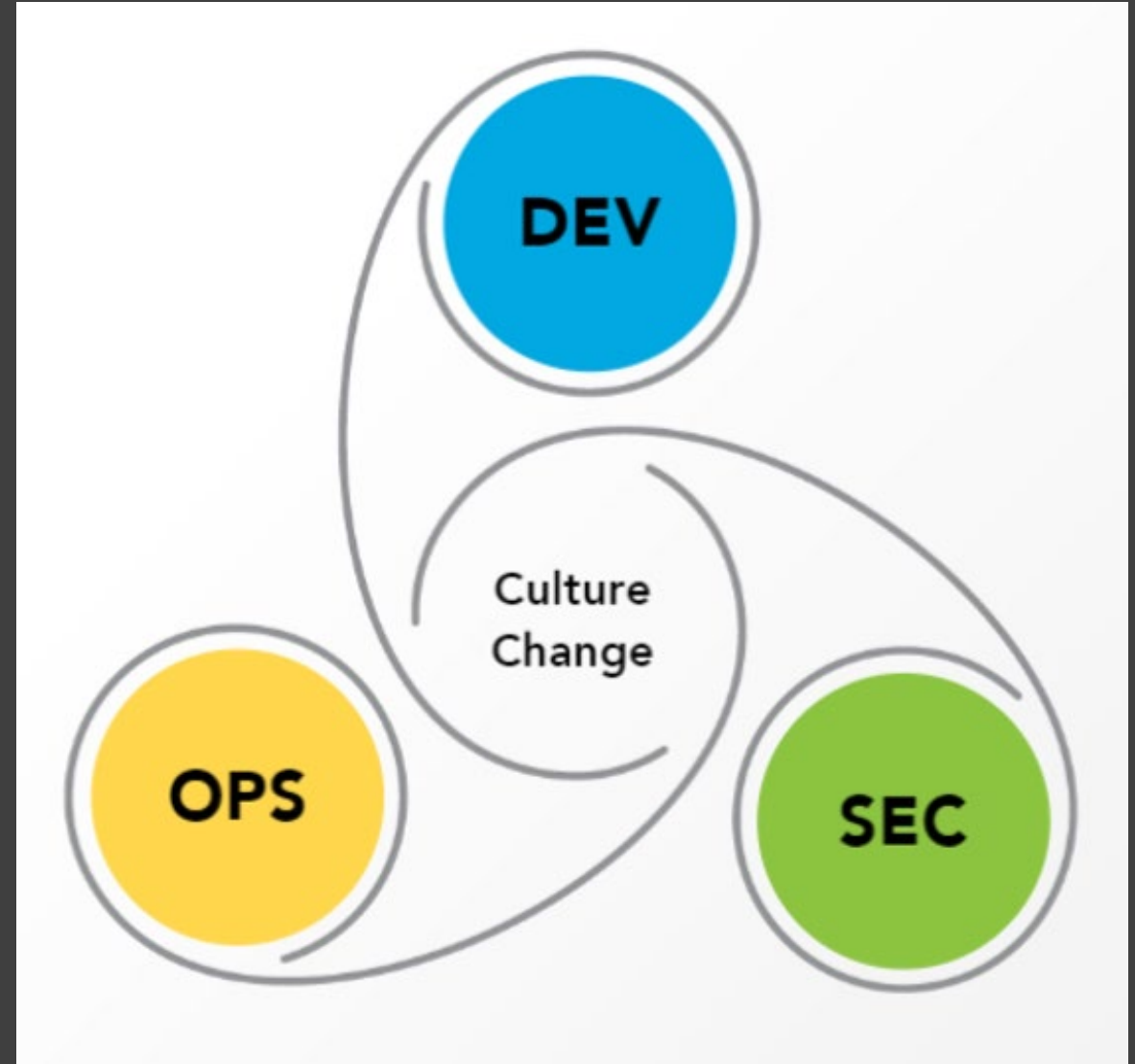
## Traditional

In the traditional view of security, operations and engineering must yield to avoid risk. A view might be that of:

- Development
- Security
- Operations

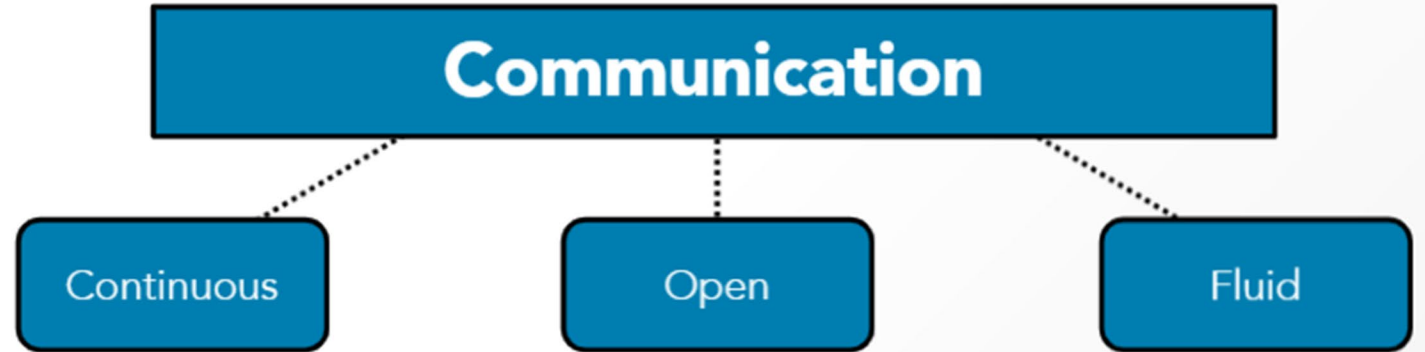
## DevSecOps

To embrace DevSecOps, security must be communicated as a core value – and as a critical enabler.




Collaboration is key!





**Communication Is Critical  
to the Cultural Change**

---



Changing behaviors &  
culture is fundamental  
to success

*Gartner Highlights Five Key Steps to Delivering an Agile  
I&O Culture; Gartner, 20 April 2015 Press Release*

# Gartner's 5-Step Approach To Cultural Challenges



Gartner Highlights Five Key Steps to Delivering an Agile I&O Culture; Gartner, 20 April 2015 Press Release



Daily Touchpoints



Wikis, Blogs & Portals



Messaging Apps



Lunch & Learn

“By 2018, 90 percent of infrastructure and operations organizations attempting to use DevOps without specifically addressing their cultural foundations will fail”

# Security Champions Facilitate a Scalable DevSecOps Program



Acting as the voice of Security



Acting as an on-site advisors



Anticipating potential design or implementation problems



Deciding when to engage the security team



Participating in code reviews and threat modeling



Troubleshooting security bugs

**AND MORE!**

“ Cultural changes come in the form of integrating teams that historically have been disparate around a **single vision**. Technical changes come with **automating** as much of the development, deployment, and operational environment as possible to more rapidly deliver **high-quality and highly secure code**. ”

# DevSecOps & Process



**Cultural change must be supported by process change**



**Security tools must be tightly integrated throughout the DevOps pipeline**



**Processes must:**

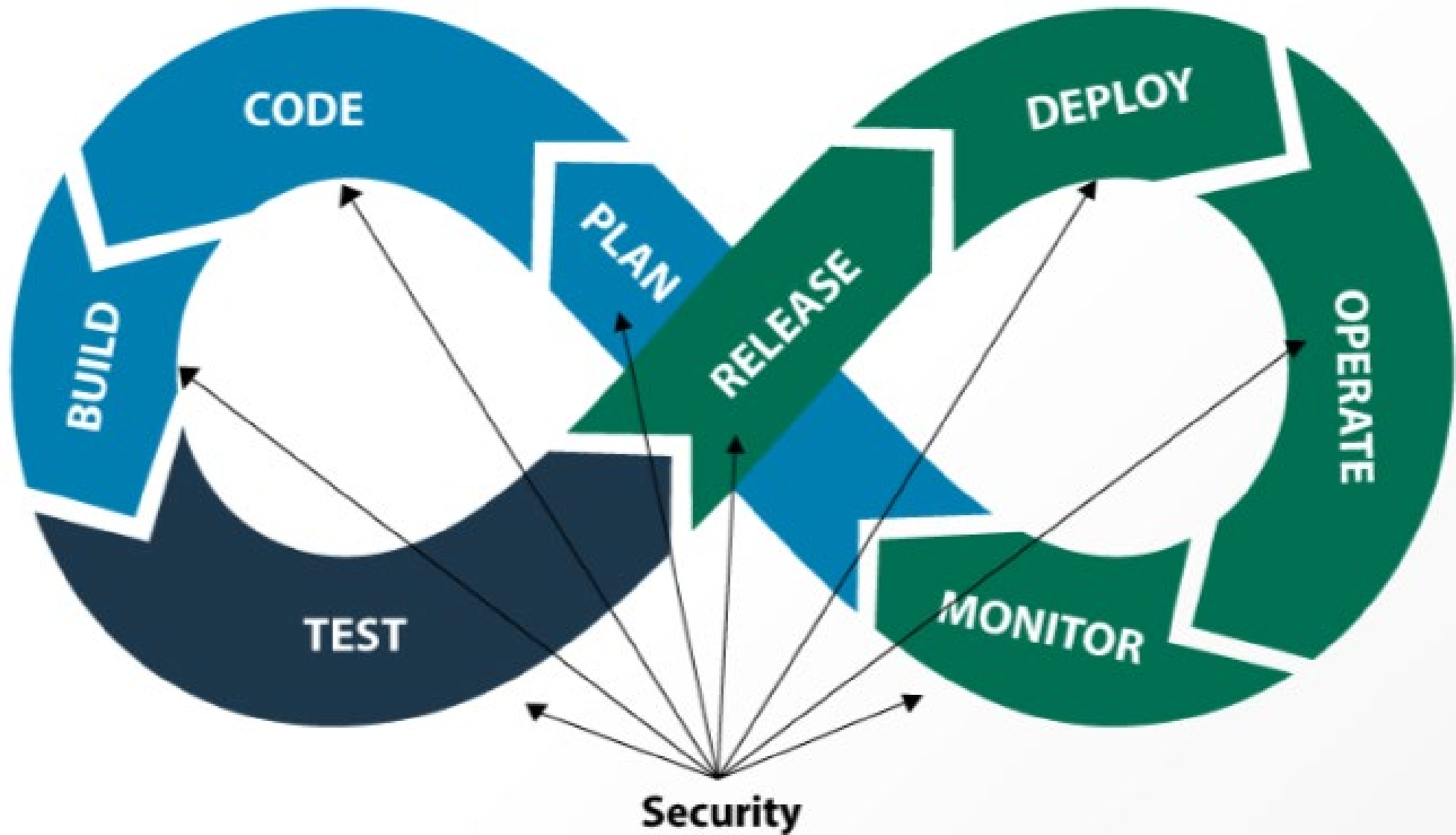
- Incorporate continuous monitoring and remediation of security defects
- Continuously test code throughout the life cycle
- Incorporate automated testing
- Support Test Driven Security (TDS)
- Support continuous & open communications



**Continual learning and improvement is key**

Recommended Reading: "Where Security Meets DevOps: Test Driven Security,"  
<https://freecontent.manning.com/where-security-meets-devops-test-driven-security/>



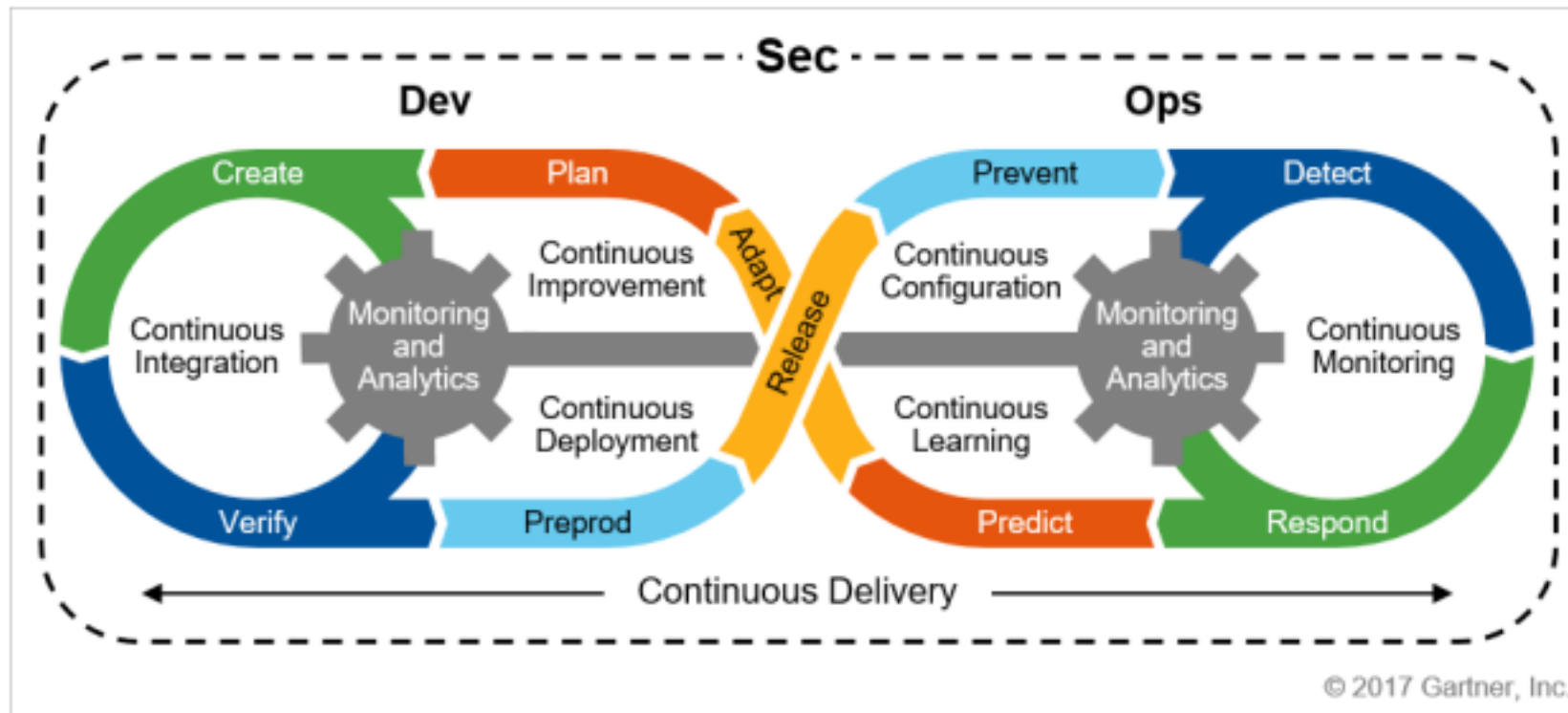


Ref: ISC2 -DevSecOps – Integrating Security into DevOps

Apply security controls in application and infrastructure layers;

Test them continuously

# Secure Development as a Continuous Improvement Process



Source: Gartner (October 2017)

# Gartner's Ten Things to Get Right....

1

Adapt your security testing tools and processes to the developers, not the other way around.

2

Quit trying to eliminate all vulnerabilities during development.

3

Focus first on identifying and removing the known critical vulnerabilities.

4

Don't expect to use traditional dynamic or static app security testing without changes.

5

Train all developers on the basics of secure coding, but don't expect them to become security experts.

6

Adopt a security champion model and implement a simple security requirements gathering tool.

7

Eliminate the use of known vulnerable components at the source.

8

Secure and apply operational discipline to automation scripts.

9

Implement strong version control on all code and components.

10

Adopt an immutable infrastructure mindset.

# 5 Principles for DevSecOps

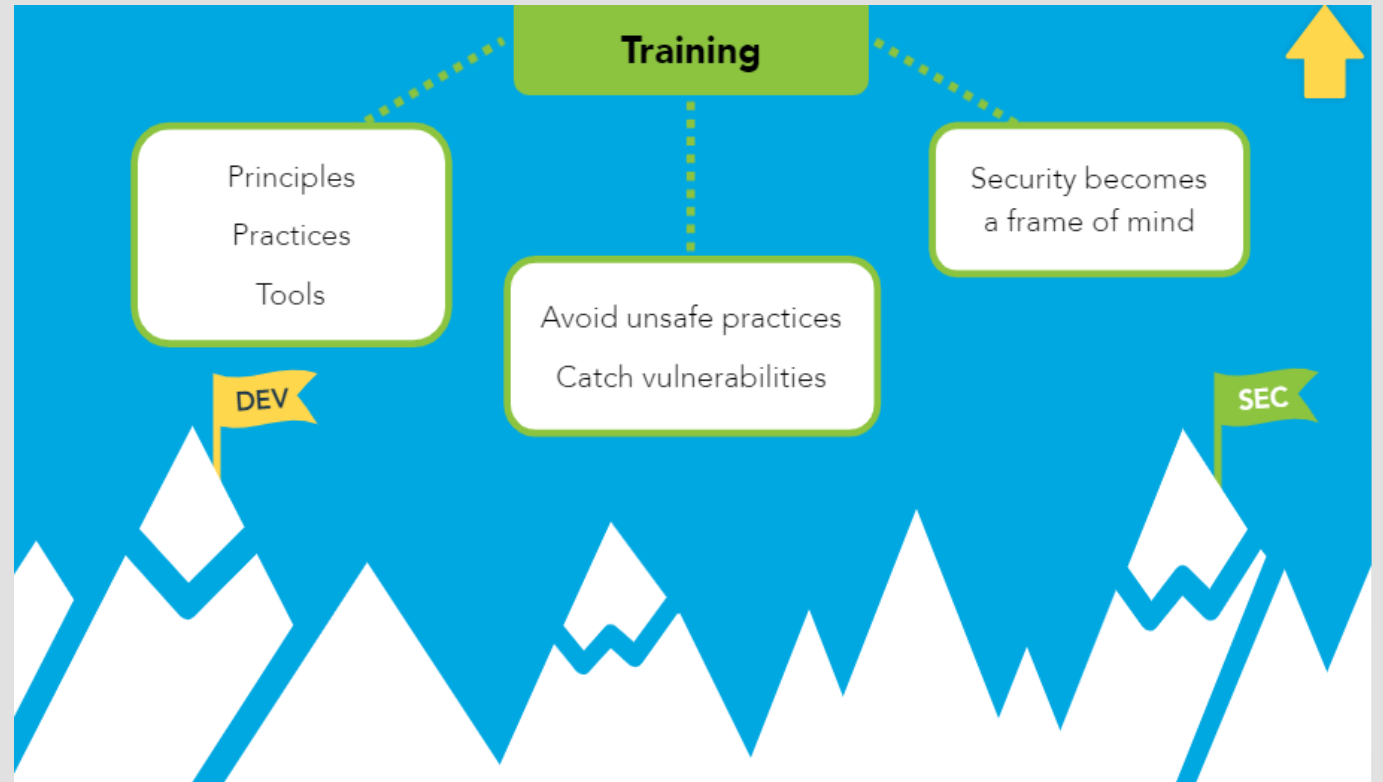
- Automate security into the process
- Integrate to fail quickly
- No false alarms
- Build security champions
- Keep operational visibility

Information extracted from "5 Principles for Securing DevOps" CA Veracode, 2018



# The Security Professional's Role

- Enable developers to find and fix security-related code defects
- Govern the use of open source components
- Implement developer training on secure coding
- Manage and report on application security policy, KPIs and metrics
- Understand the requirements for security testing solutions in a DevSecOps environment
- Create developer security champions



Recommended reading: "The Security Professional's Role in a DevSecOps World," <https://info.veracode.com/guide-the-security-professionals-role-in-devops-world.html>



# DevSecOps Tools

# DevSecOps Tools – The Third Leg of the Stool

Automated testing is key to driving the DevOps pipeline

As noted - Security tools must be tightly integrated throughout the DevOps pipeline

Testing using tools should be metric driven a few key metrics include:

- Availability: Amount of uptime/downtime in a given time period, in accordance with the SLA.
- Change Failure: Percentage of production deployments that failed.
- Change Lead Time: Time between a code commit and production deployment of that code.
- Mean Time to Failure (MTTF): Time that a system is online between outages or failures.
- Mean Time to Recovery (MTTR): Time between a failed production deployment to full restoration of production operations.
- Number of False Positives: The number of mistakenly flagged vulnerabilities for an application.
- ISC2 list in appendix.

DevSecOps  
Tools Drive  
the DevOps  
Pipeline Via  
Logging

## Logging pipeline

Analyze usage

Analyze security incidents



DevOps teams may not know  
how to identify security  
breaches, hacking attempts

Log management tool

Reading & parsing logs

Distinguishing  
unauthorized activity



A person is seen from behind, sitting at a desk in a dark room. They are looking at a large computer monitor displaying code or data. A laptop is open on the desk in front of them. The room is dimly lit, with light coming from the screens and a window in the background showing city lights at night. The word "Summary" is overlaid in white text on the left side of the image.

# Summary

# The Case for DevSecOps

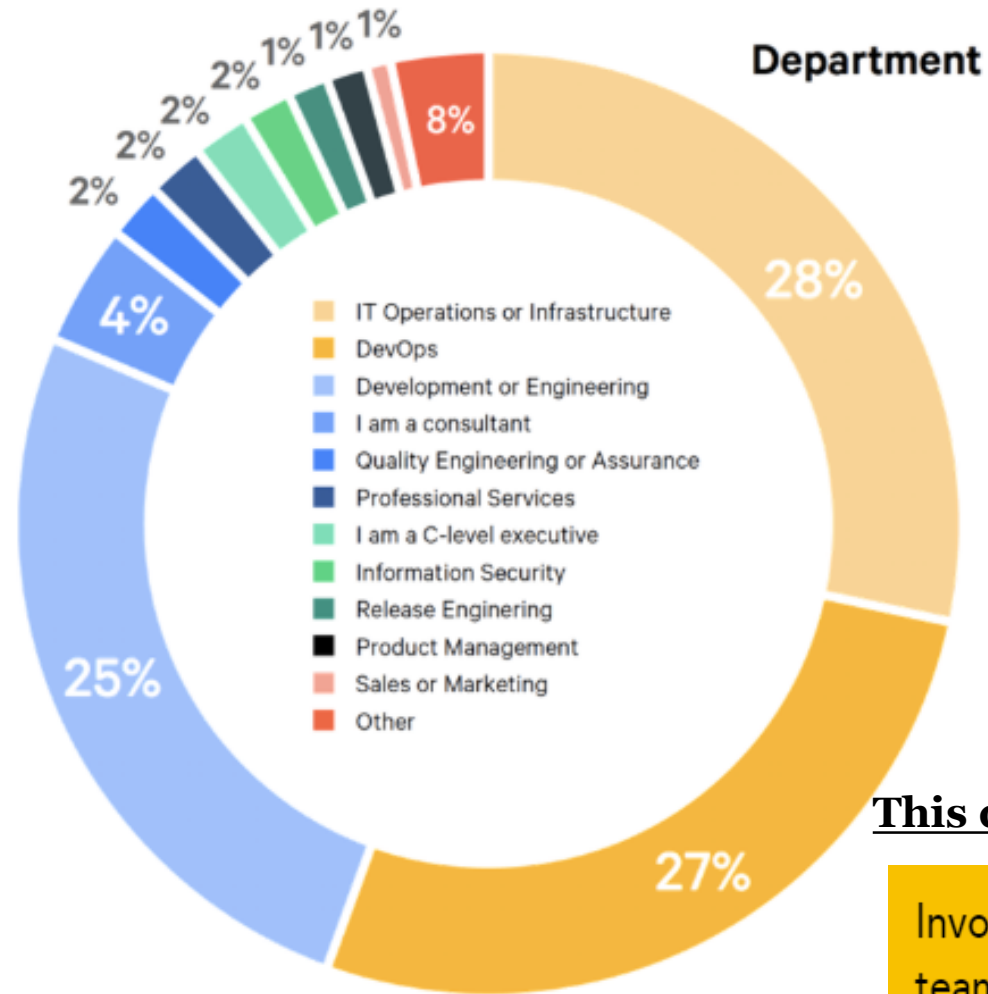
2014 — 16% DevOps teams

2017 — 27% DevOps teams

2018 — 29% DevOps teams

High performers spent 50% less time remediating security issues

*State of DevOps Report 2018 Presented by Puppet and Splunk (released during the development of this course) indicates 29%*



**This drives the need to:**

Involve security and quality teams in the development process early and often.

# A Security Strategy for Implementing DevSecOps

---

## Tools & Frameworks

---

- ✓ Culture
  - ✓ Process
  - ✓ Technology
- 

## Keys to Successful Implementation

- Culture of Collaboration and Contribution
  - Everyone has something to offer
  - Everyone is responsible for security
  - Goal = safely distributing security decisions
- Process – signification changes to existing processes
  - Need mechanisms for communications, measurement, reporting
  - Need to establish a group including Security, Development and Operations
    - **This group is responsible for end-to-end security:**
      - **App development**
      - **Implementing changes**
      - **A continuous loop – CI/CD**
- Tools – required to automate processes for:
  - Managing code repositories
  - Testing – attacking surface analysis, threat modeling, penn & fuzz testing, *etc.*



Q & A

# Thank You



**Eddie McAndrew**  
COO  
AIS Network  
(804) 239-5185

Email:  
[eddie.mcandrew@aisn.net](mailto:eddie.mcandrew@aisn.net)



**Barry Davis**  
CISSO  
Virginia Dept. of Social Services  
(804) 726-7153

Email:  
[barry.davis@dss.virginia.gov](mailto:barry.davis@dss.virginia.gov)

# Appendix 1 – ISC2 DevSecOps KPIs



## Key Performance Indicators

### TERMS: DEFINITION

---

**Availability:** Amount of uptime/downtime in a given time period, in accordance with the SLA.

**Change Failure:** Percentage of production deployments that failed.

**Change Lead Time:** Time between a code commit and production deployment of that code.

**Change Volume:** Number of user stories deployed in a given time frame.

**Customer Issue Resolution Time:** Mean time to resolve a customer-reported issue.

**Customer Issue Volume:** Number of issues reported by customers in a given time period.

**Defect Burn Rate:** Amount of time to fix vulnerabilities in an application.

**Defect Density:** The number of bugs identified divided by the codebase of an application.

**Deployment Frequency:** Number of deployments to production in a given time frame.

**Logging Availability:** Amount of uptime/downtime of the logging pipeline in a given time period.

**Mean Time Between Failures (MTBF):** The amount of time that elapses between one failure and the next. Mathematically, this is the sum of MTTF and MTTR, the total time required for a device to fail and that failure to be repaired.

**Mean Time to Failure (MTTF):** Time that a system is online between outages or failures.

**Mean Time to Recovery (MTTR):** Time between a failed production deployment to full restoration of production operations.

**Number of False Positives:** The number of mistakenly flagged vulnerabilities for an application.

**Number of Functional/Acceptance Tests:** Number of automated functional or acceptance tests for an application.

**Number of Passed/Failed Security Tests:** Number of automated security tests for an application.



# Appendix 2

## – ISC2



### Security Professional's role

Today, modern application security programs feature centralized governance by security, but testing and fixing are owned by development in an automated fashion throughout the build process. In this approach, security owns setting policies, tracking KPIs, and providing security coaching to developers.

In addition, security is responsible for providing developers with support in integrating scalable tools into their SDLC. Developers own testing applications in their development environment, fixing flaws to pass policy, and continuing to build code.

In this process, security-related defects are just another bug during the build process, and developers have the tools and guidance needed to fix them. At the same time, security can govern the program to make sure KPIs and policies are met.

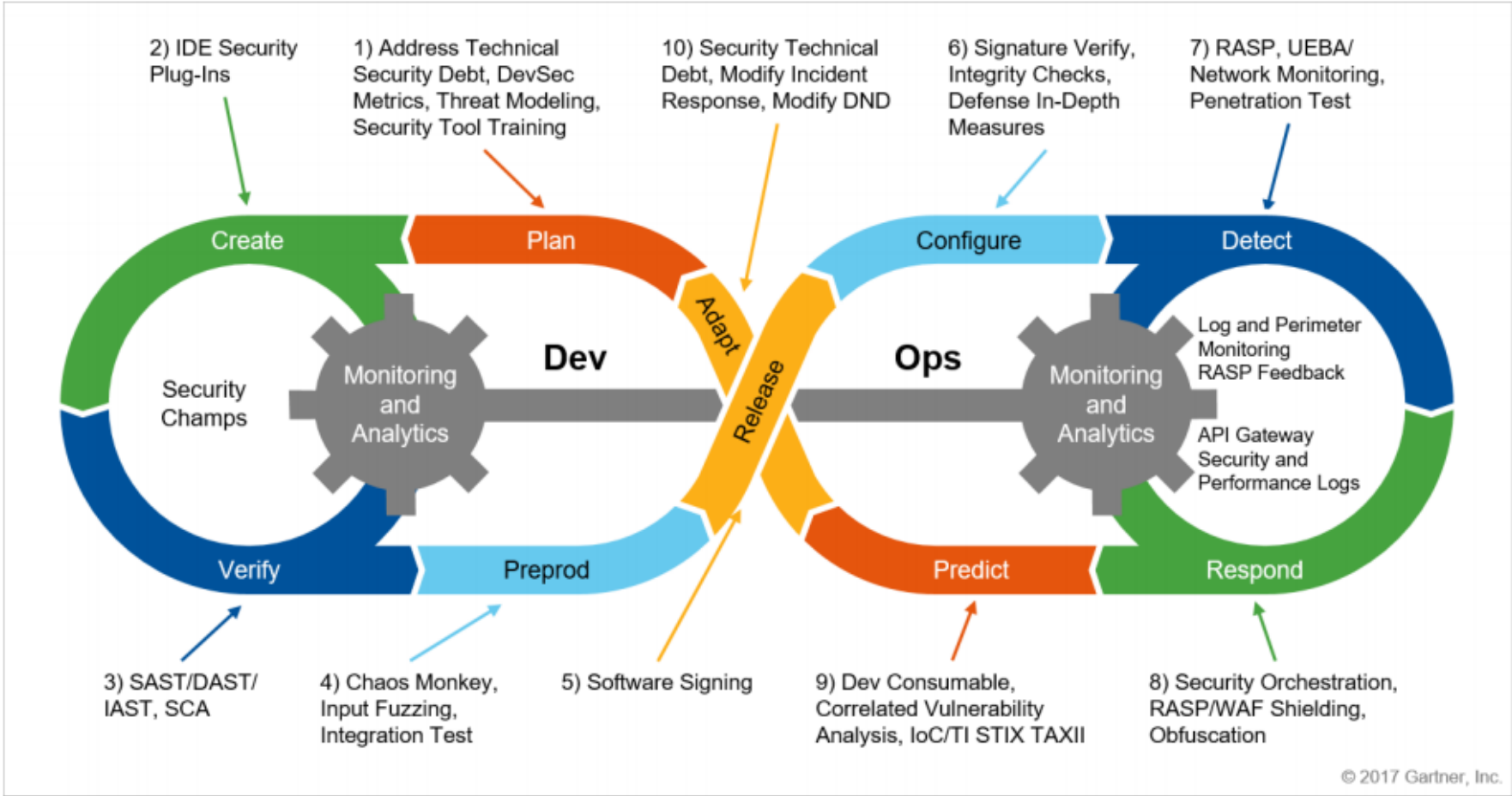
In this realm, security professionals will have new responsibilities and new skill requirements.

|  | NEW SKILL REQUIREMENTS   |
|--|--|
| Enable developers to find and fix security-related code defects  | Ability to provide remediation coaching and guidance on security-related code defects          |
| Govern the use of open source components   | Basic understanding of application development and why and how third-party components are used |
| Implement developer training on secure coding  | Understanding of the basics of software development  |
| Manage and report on application security policy, KPIs and metrics   | The ability to measure meaningful metrics at each point in the SDLC process                    |
| Understand the requirements for security testing solutions in a DevSecOps environment — including the need for immediacy and accuracy of results to avoid impacting the delivery cycle — and enable dev to use these solutions | Basic understanding of application development and why and how third-party components are used |
| Create developer security champions  | Be empathetic and consultative   |

Ref: VERACODE GUIDE - THE SECURITY PROFESSIONAL'S ROLE in a DevSecOps World



# DevSecOps Tooling



Source: Gartner (November 2017)