From DevOps To DevSecOps

[Karan Sharma]

Thank You to Our Sponsors and Hosts!

OWASP NEW ZEALAND
owasp.org.nz

AppSec NZ
appsec.org.nz

AUT UNIVERSITY
TE WĀNANGA ARONUI O TAMAKI MAKAU RAU

DEFEND™

fastly®

QUANTUM SECURITY

DATACOM

aura INFORMATION SECURITY | POWERED BY KORDIA

IriusRisk

snyk

dta
Defence Technology Agency

planit an NRI company

CyberCX

tesserent

FIGHTING FOR FAIR FOR KIWI BUSINESS

safeadvisory.

Without them, this Conference couldn't happen.

$ whoami

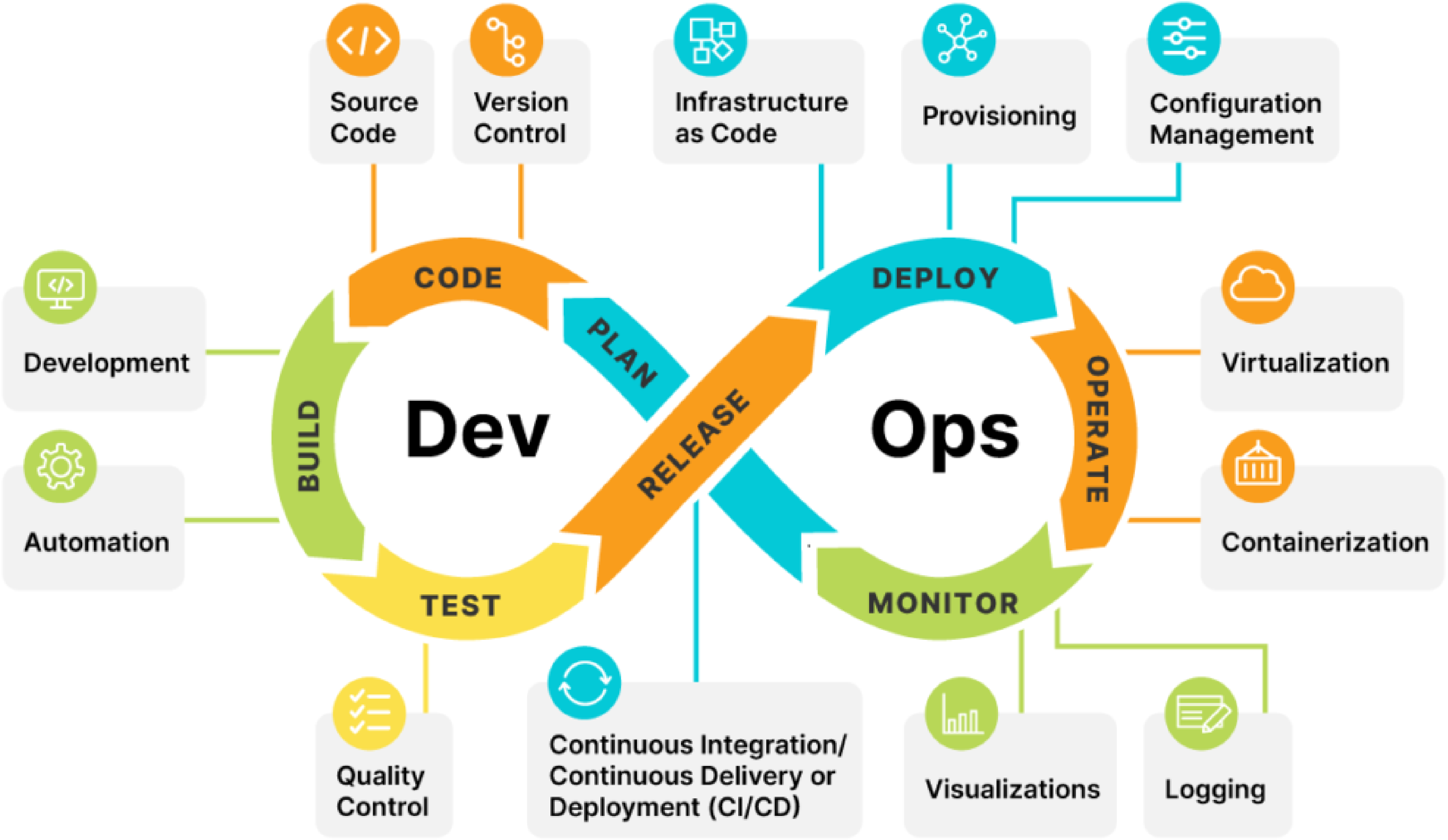
- Founder @ Wise Fox Security
- CDP | OSWE | OSCP | eWPTX certified
- Passion - Web/Mobile | DevSecOps | Code Auditing
- Twitter - @W1S3F0X
- YouTube Channel - Wise Fox Security

Agenda

- What & Why DevSecOps
- Challenges while implementing DevSecOps
- How to overcome such challenges
- DevSecOps Pipeline Walkthrough
- Best Practices for implementing DevSecOps

What is 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.
- Under a DevOps model, development and operations teams are no longer siloed.
- DevOps encourages automation, CI & CD, so changes to the software can be quickly and reliably deployed, tested, and released.
- It helps teams work together efficiently and helps deliver better quality software to users more frequently.



https://productcoalition.com/12-top-devops-best-practices-for-a-successful-transition-in-2023-b73b54014d0d

What is DevSecOps?

- DevSecOps stands for Development, Security, and Operations.

- It is an approach that integrates security practices into the software development and delivery processes.

- DevSecOps aims to shift security from being an afterthought to an integral part of the SDLC.

Plan and Develop

- ☐ Threat modelling
- ☐ IDE Security plugins
- ☐ Pre-commit hooks
- ☐ Secure coding standards
- ☐ Peer review

Commit the code

- ☐ Static application security testing
- ☐ Security unit and functional tests
- ☐ Dependency management
- ☐ Secure pipelines

Build and test

- ☐ Dynamic application security testing
- ☐ Cloud configuration validation
- ☐ Infrastructure scanning
- ☐ Security acceptance testing

Go to production

- ☐ Security smoke tests
- ☐ Configuration checks
- ☐ Live Site Penetration testing

Operate

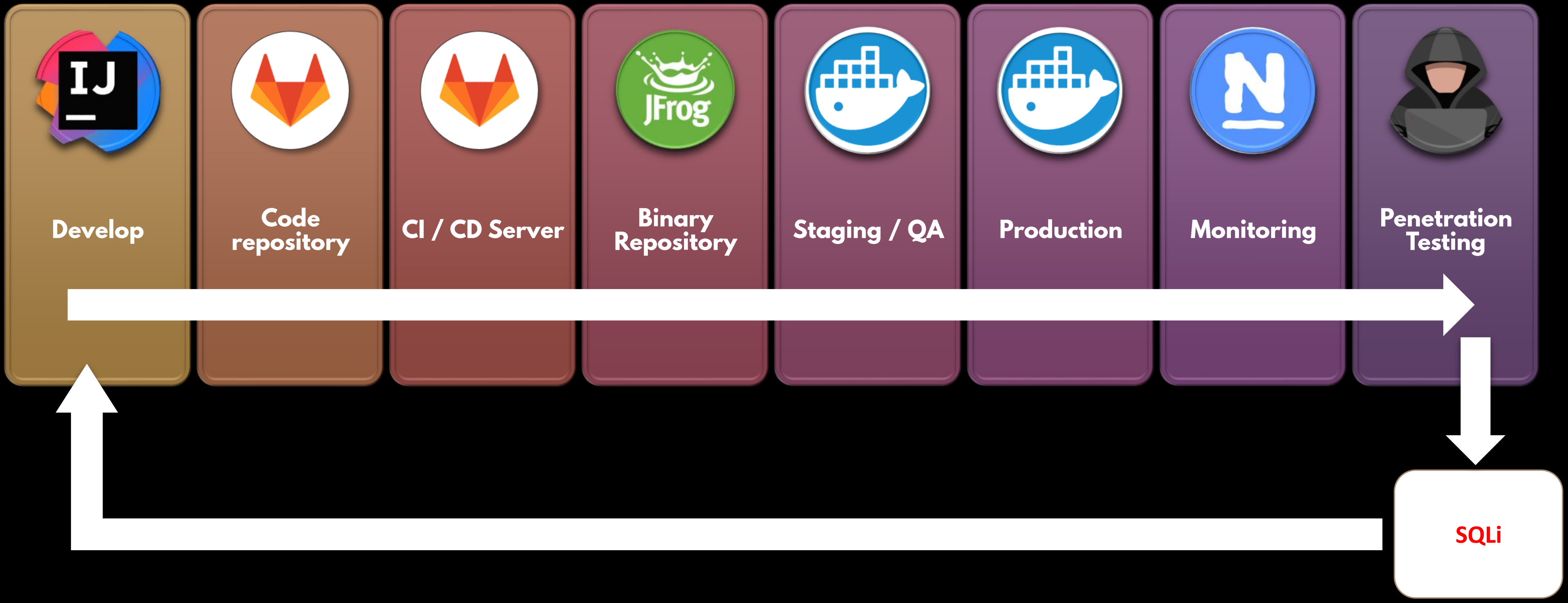
- ☐ Continuous monitoring
- ☐ Threat intelligence
- ☐ Penetration testing
- ☐ Blameless postmortems

https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/secure/media/devsecops-controls.png

DevSecOps Benefits

- Faster vulnerability identification and remediation for robust security.
- Continuous security assessments through automated security testing.
- Enhanced collaboration, breaking down team silos.
- Cost-effective since security issues are addressed early on.
- Demonstrate commitment to data protection and standards by incorporating security measures in the development process.

Traditional Approach



Develop

Code repository

CI / CD Server

Binary Repository

Staging / QA

Production

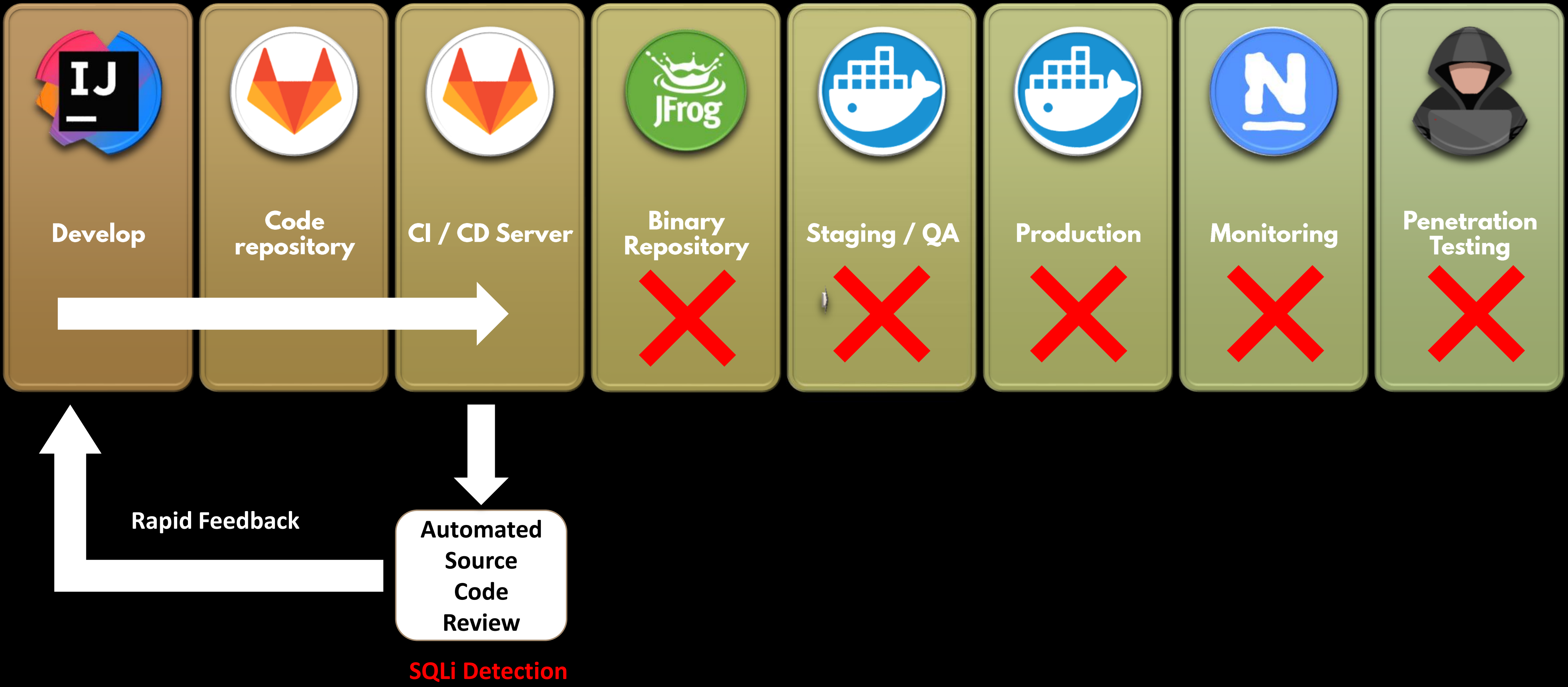
Monitoring

Penetration Testing

SQLi

https://notsosecure.com/achieving-devsecops-open-source-tools

The Shift Left Approach



https://notsosecure.com/achieving-devsecops-open-source-tools

DevSecOps

- **Lack of Budget**
  - Limited financial resources can hinder the implementation of DevSecOps practices.
  - It can hinder the acquisition of necessary security tools and technologies.
  - Organization may struggle to invest in training and development programs for security awareness and upskilling.

- **Cultural Shift**
  - Siloed DevOps and Security teams.
  - Development teams might be accustomed to focusing solely on delivering features on tight deadlines.
  - On the other hand, security teams may prioritize risk mitigation over delivery.
  - No collaboration between security and other teams.
  - Security teams lack visibility.

- **Skillset and Knowledge Gap**
    - Lack of knowledge or understanding around secure coding practices or security principles in general.
    - Security and Operations teams may not be familiar with both infrastructure and software development environments.
    - Organization lacks resources, guides or standards relating to security.
    - Lack of common platform or a programme to share knowledge.

- **Toolchain Integration**
  - Existing security tooling may not be compatible with the DevOps processes or practices.
  - Unaudited open-source tools already in use.
  - Tools selection criteria or process is not well defined.
  - DevOps teams are not comfortable using tools selected by the security teams.
  - Friction over pipeline failures due to security tooling.

- **Roles & Responsibilities**
  - Developers assume the security team is responsible for security and risk mitigation which leads to misunderstanding and gaps in security practices.
  - In reality, security team is there to:
    - Establish security policies and Guardrails
    - Guide developers in understanding security requirements
    - Provide security best practices
    - Security training
    - Advisory role

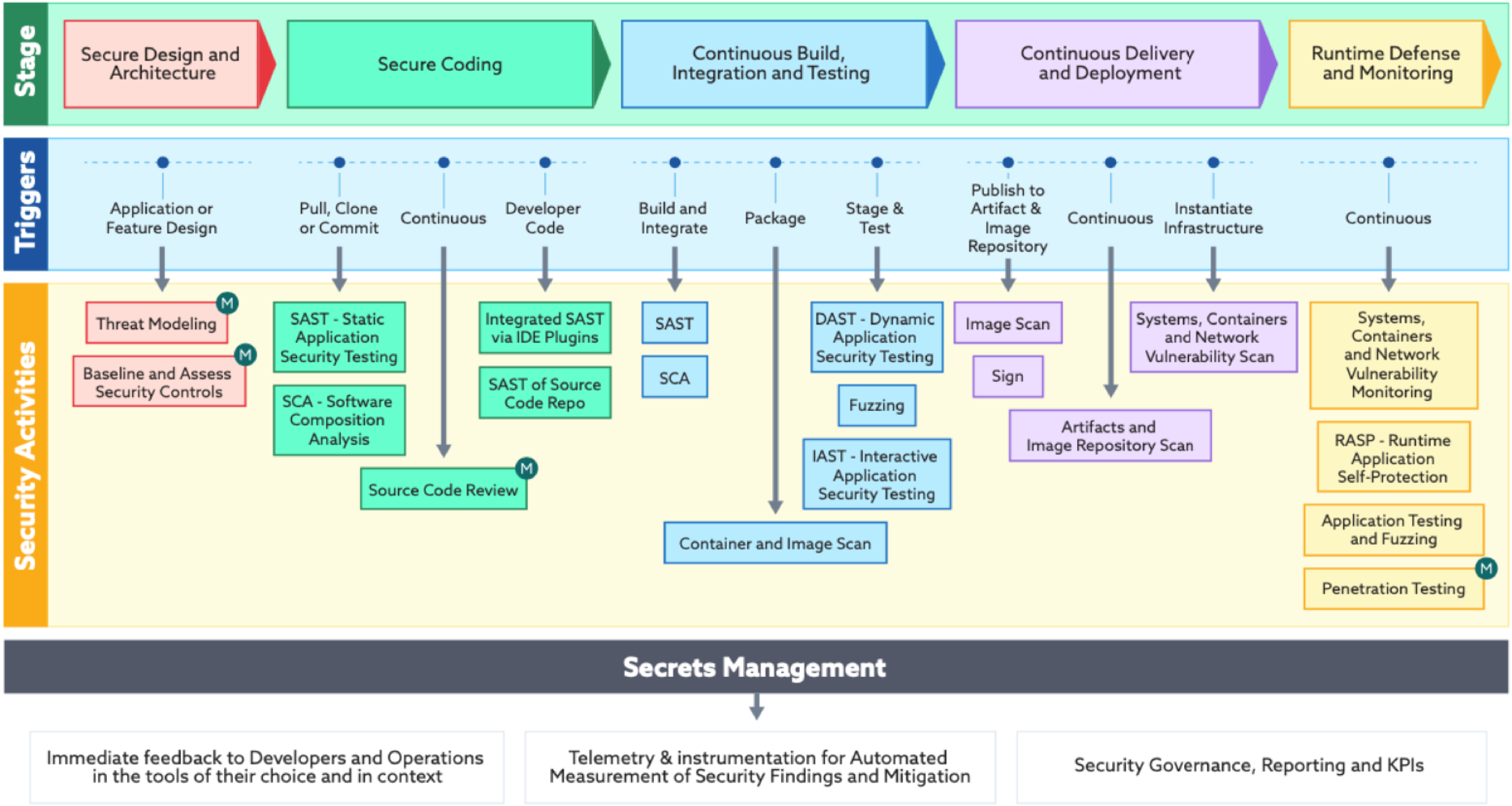
SUCCESS
AHEAD

Start Small

Start with Open-Source Tooling

Security Activity	Open-Source Tooling
Threat Model	MS Threat-Modeling tool, OWASP Threat Dragon
SAST	Brakeman, Bandit, Snyk, SonarQube
SCA	Retire.js, Safety, OpenSCA
Secrets Scanning	Trufflehog, Detect-secret, Gitleaks
DAST	OWASP ZAP, Nikto, StackHawk, Arachni
Container Scanning	Clair, Grype, Trivy
IaC SAST	Checkov, tfsec, Terrascan
Docker file Scanning	Checkov, Docker scan
SBOM Scanning	Dependency-check, Syft, Grype
API Security Testing	Astra, Postman

Create a DevSecOps Framework



Stage

Secure Design and Architecture

Secure Coding

Continuous Build, Integration and Testing

Continuous Delivery and Deployment

Runtime Defense and Monitoring

Triggers

Application or Feature Design

Pull, Clone or Commit

Continuous

Developer Code

Build and Integrate

Package

Stage & Test

Publish to Artifact & Image Repository

Continuous

Instantiate Infrastructure

Continuous

Security Activities

Threat Modeling M

Baseline and Assess Security Controls M

SAST - Static Application Security Testing

SCA - Software Composition Analysis

Source Code Review M

Integrated SAST via IDE Plugins

SAST of Source Code Repo

SAST

SCA

Container and Image Scan

DAST - Dynamic Application Security Testing

Fuzzing

IAST - Interactive Application Security Testing

Image Scan

Sign

Artifacts and Image Repository Scan

Systems, Containers and Network Vulnerability Scan

Systems, Containers and Network Vulnerability Monitoring

RASP - Runtime Application Self-Protection

Application Testing and Fuzzing

Penetration Testing M

Secrets Management

Immediate feedback to Developers and Operations in the tools of their choice and in context

Telemetry & instrumentation for Automated Measurement of Security Findings and Mitigation

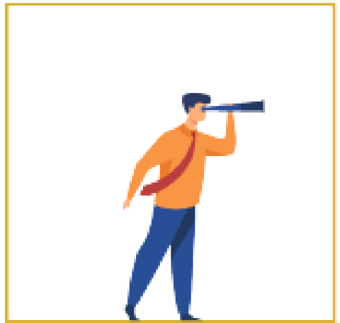
Security Governance, Reporting and KPIs

Image Source - https://devsecops.pagerduty.com/secure_sdlc/

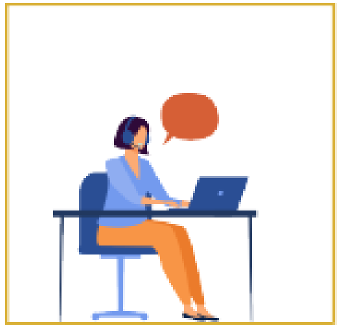
Establish Security Champions Programme

THE SECURITY CHAMPIONS MANIFESTO

Be Passionate About Security



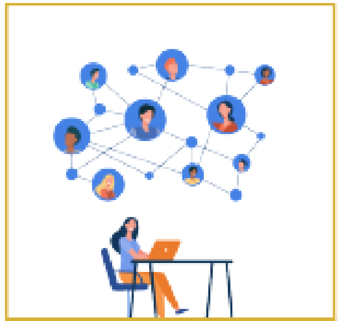
Start With a Clear Vision



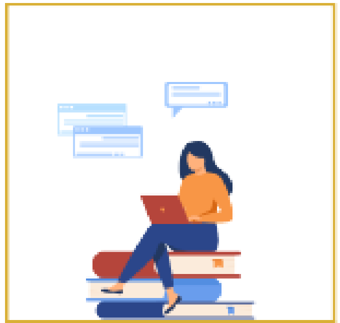
Secure Management Support

Nominate a Dedicated Captain

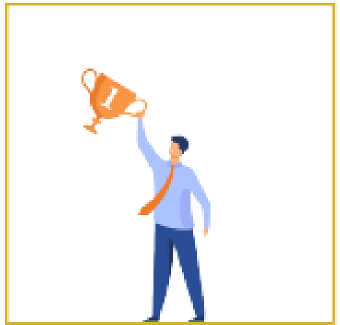
Trust Your Champions



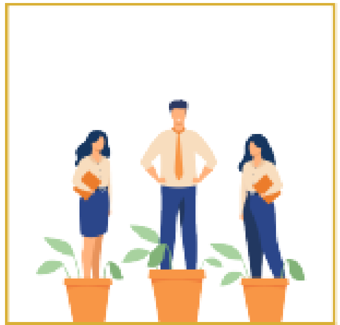
Create a Community



Promote Knowledge Sharing



Reward Responsibility



Invest in Your Champions

Anticipate Personnel Changes

OWASP SECURITY CHAMPIONS GUIDE

Upskill your existing resources

Cross-Functional Collaboration

Risk-based approach for pipeline release

- Know your DevSecOps maturity
- What are your Crown-Jewels that you are trying to protect
- Categories them

Platinum
Gold
Silver
Bronze

- Define risk-appetite for each of the category

Platinum
Gold
Silver
Bronze

CRITICAL
HIGH
MEDIUM

CRITICAL
HIGH

CRITICAL

CRITICAL

- Tune your tools based on the risk-appetite for pipeline failures

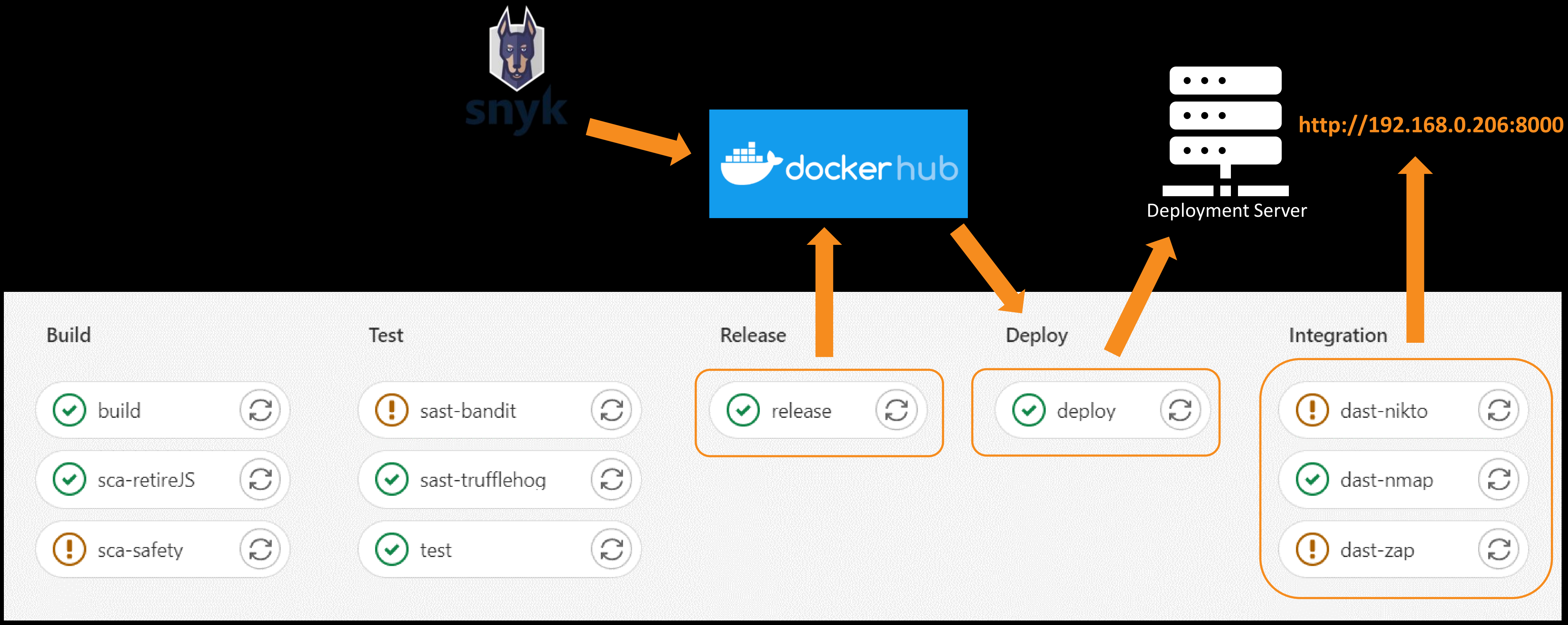
Measure your success

- **Mean Time to Detect (MTTD)**
  - *Time it takes to detect security vulnerabilities in dev or prod environments.*
- **Mean Time to Remediate (MTTR)**
  - *Time it takes to remediate identified security issues.*
- **Vulnerability Density**
  - *Number of vulnerabilities identified per unit of code or application.*
- **Automated Security Test Coverage**
  - *Percentage of security tests automated.*
- **Security Tooling Coverage**
  - *How many applications/services or products using security tooling.*
- **Number of False Positives**
  - *Number of false/positive results in the security tooling.*

https://chainwide.io/blog/post/2247798/how-to-measure-devsecops-success

Engage External Resources

Pipeline Walkthrough



Deployment Server

http://192.168.0.206:8000

Build	Test	Release	Deploy	Integration
✓ build	! sast-bandit	✓ release	✓ deploy	! dast-nikto
✓ sca-retireJS	✓ sast-trufflehog			✓ dast-nmap
! sca-safety	✓ test			! dast-zap

Helpful Tips

- Automate, automate, automate
- Shorter scan intervals
- KPIs and Metrics
- Don’t fail the pipelines when your security maturity is low
- Tune your tools as you go
- Collaborate more
- Don’t forget about the security activities on the right
- Train Security Champions to do Threat-Modelling
- Automate APIs Authz security tests ☺

ANY QUESTIONS?
memecrunch.com

THANK
YOU