

DevSecOps: The Big Picture

UNDERSTANDING DEVSECOPS CONCEPTS



Richard Harpur

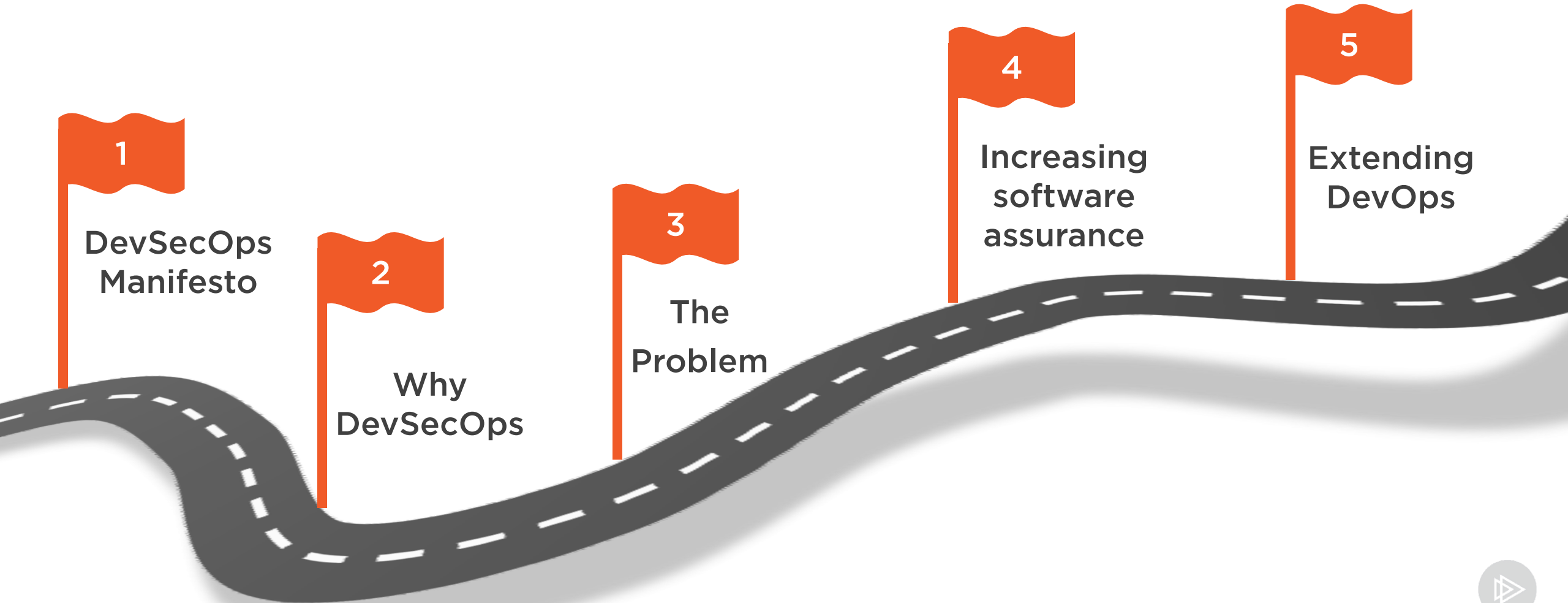
INFORMATION SECURITY PROFESSIONAL, CISM

@rharpur

www.richardharpur.com



Welcome to Our DevSecOps Journey



Overview



DevSecOps manifesto – the north star

Why DevSecOps

Problem being solved

How DevOps is extended with DevSecOps

DevSecOps increases software assurance





The DevSecOps Concept

How to incorporate security within agile and DevOps practices.



DevSecOps

The purpose and intent of DevSecOps is to build on the mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

Shannon Lietz

Source: <https://www.devsecops.org/blog/2015/2/15/what-is-devsecops>

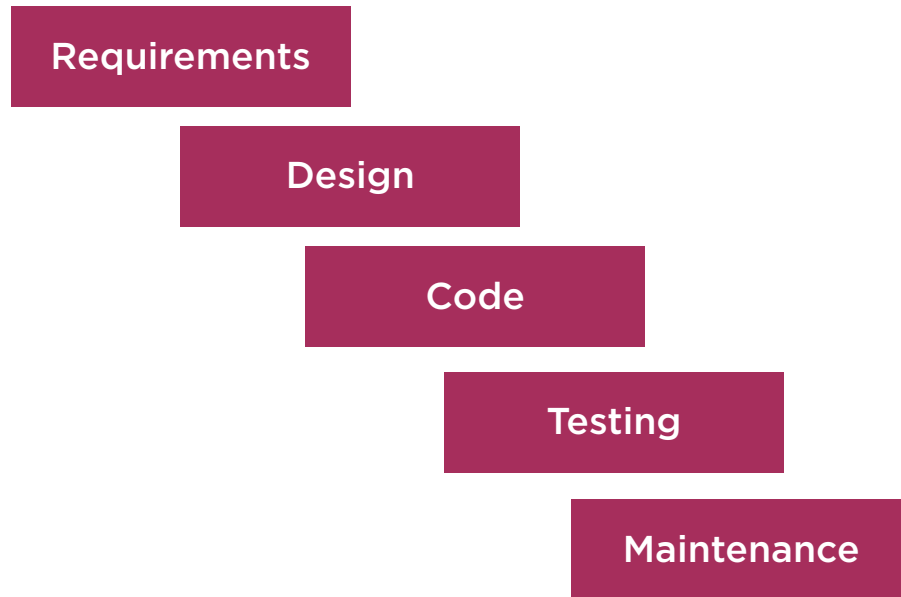


DevSecOps Manifesto

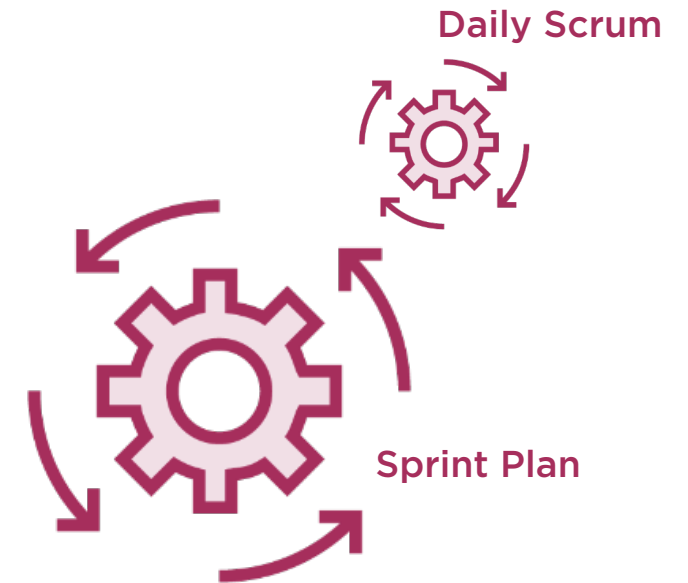
Leaning in over Always Saying “No”
Data & Security Science over Fear, Uncertainty and Doubt
Open Contribution & Collaboration over Security-Only Requirements
Consumable Security Services with APIs over Mandated Security Controls & Paperwork
Business Driven Security Scores over Rubber Stamp Security
Red & Blue Team Exploit Testing over Relying on Scans & Theoretical Vulnerabilities
24x7 Proactive Security Monitoring over Reacting after being Informed of an Incident
Shared Threat Intelligence over Keeping Info to Ourselves
Compliance Operations over Clipboards & Checklists



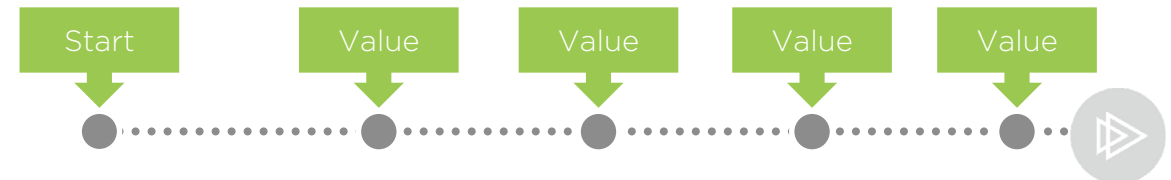
Software Development Evolution



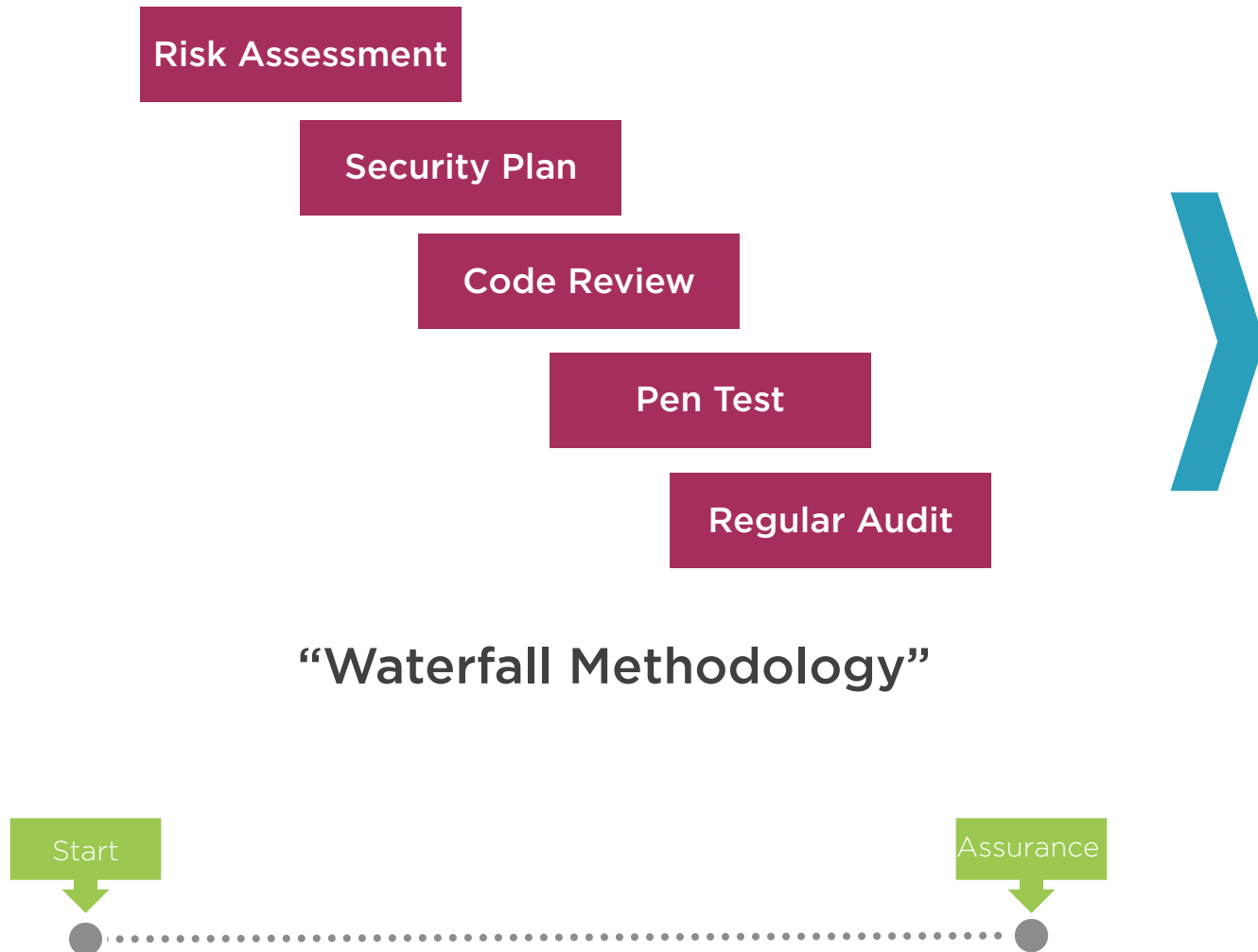
Waterfall Methodology



Agile Methodology



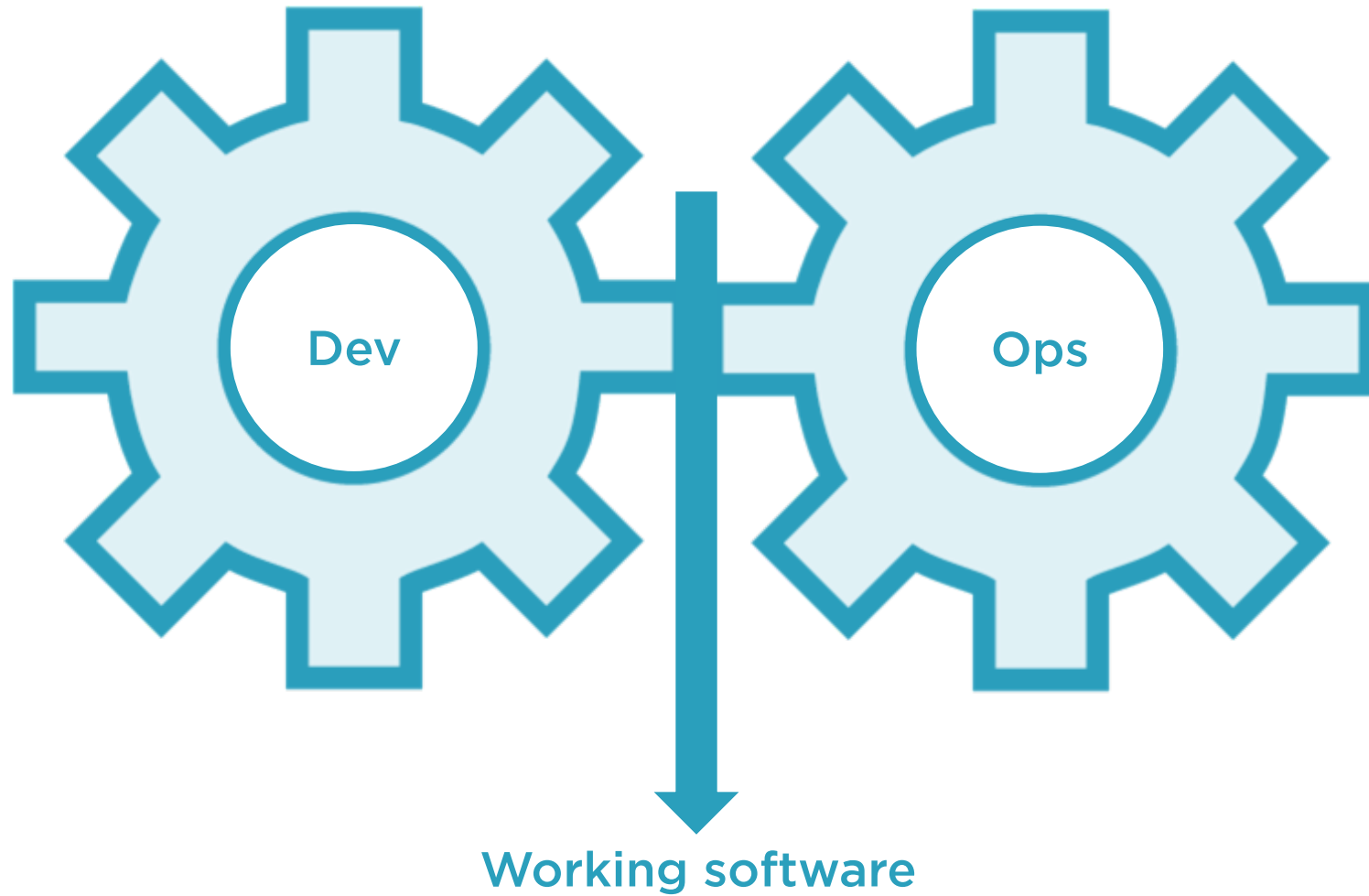
Information Security Friction



- Security as an after-thought
- Security “sign off” delays project
- Issues identified late in project
- Once-off point in time assessment
- Cost of re-testing is very high
- Security is too slow
- Not enough skills available to be secure
- Ratio of Security Experts to Dev Experts is very low



Is This Similar to DevOps?



Development and Operations Friction

Development

Improvement Focused – Frequent Change

Developers own the change

Planning Flexibility

Once released, move on to next version

Operations

Stability Focused – Limit Change

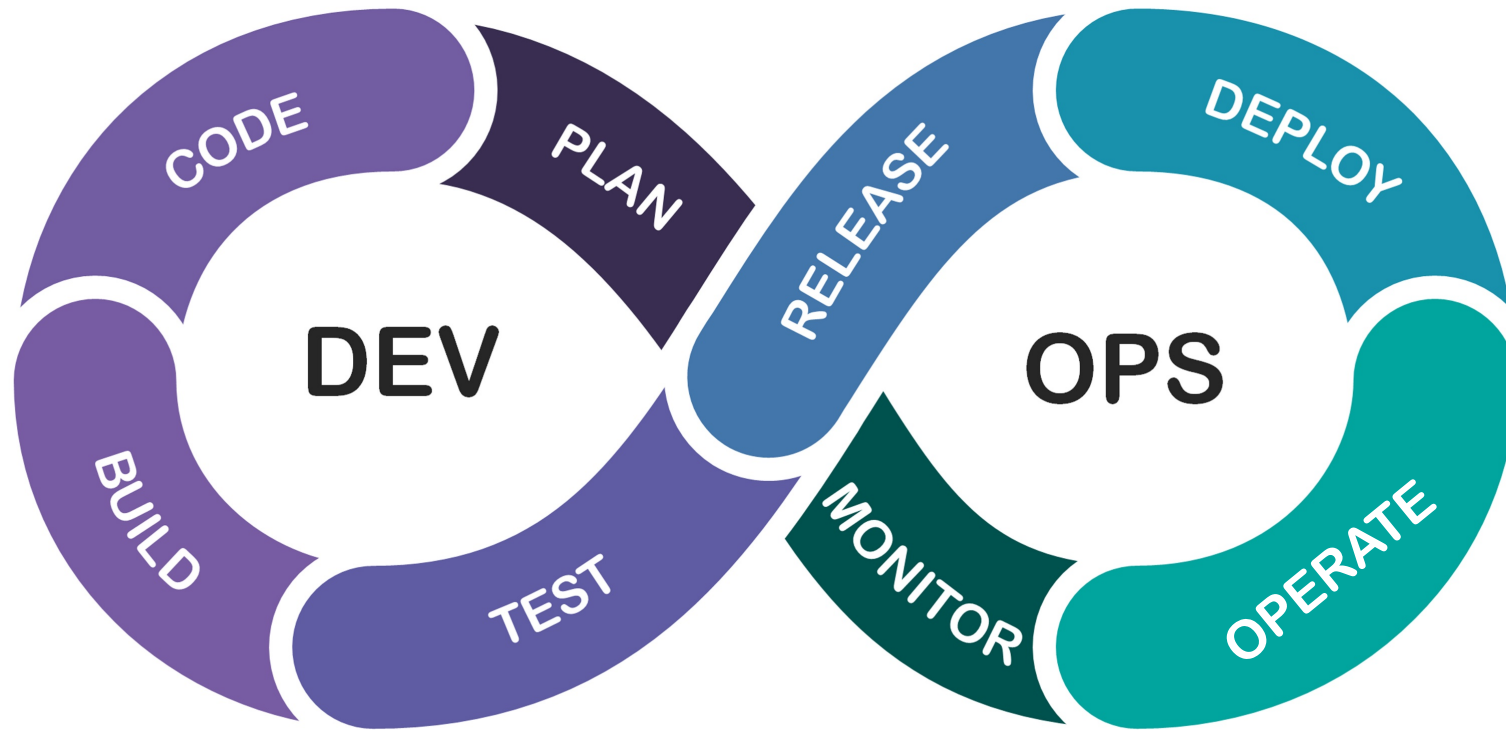
Engineers follow procedures (SOPs)

Rigid Change Advisory Board

After release, have constant monitoring,
and 24x7 response support



Merge Dev and Ops to Remove Friction



An Example of “Best Practice”

PCI DSS V 3.2.1

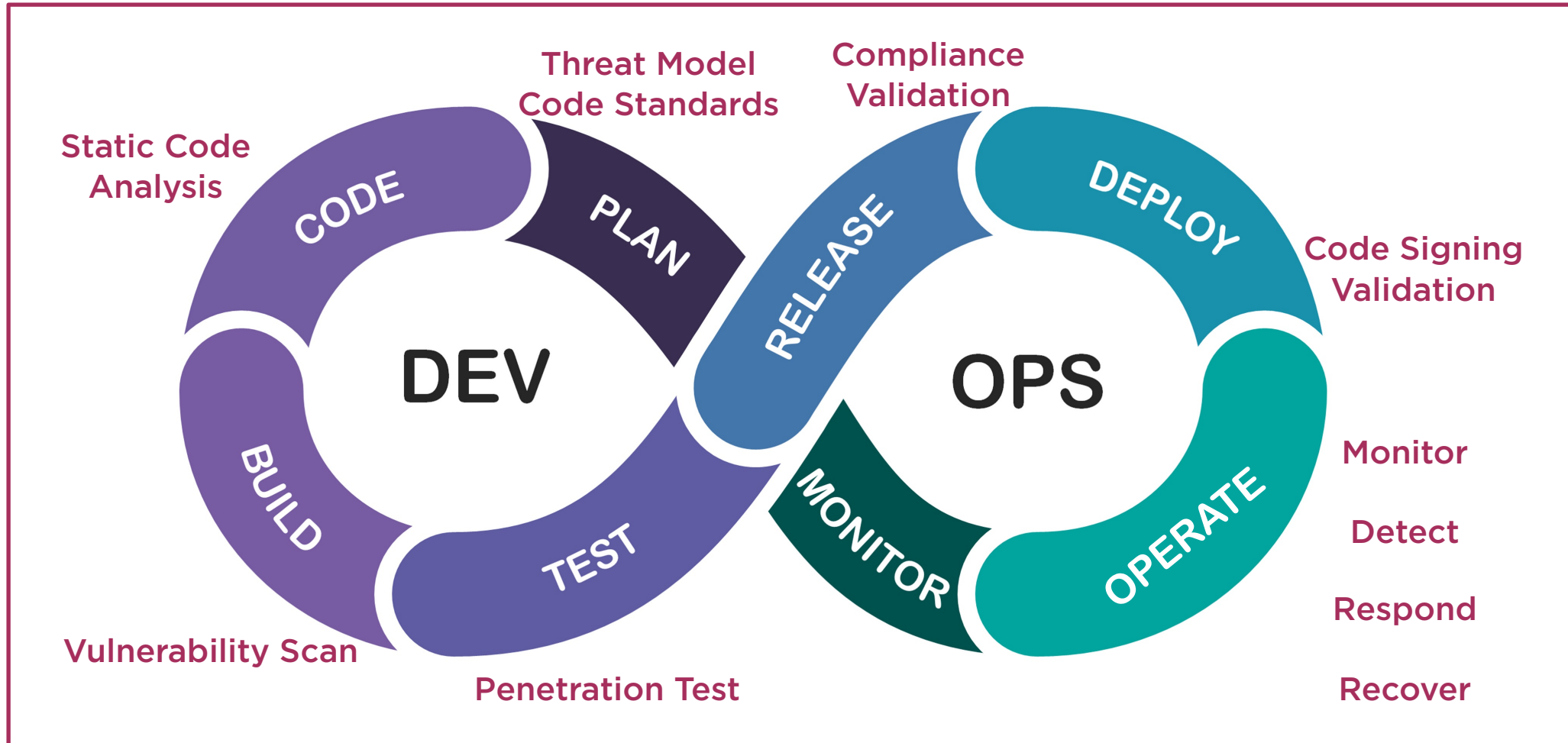
11.2 Run internal and external network vulnerability scans at least quarterly and after any significant change in the network

Initiate a Vulnerability Scan once per quarter; have a human review the results and re-test to ensure “High Risk” vulnerabilities were addressed

Undertake vulnerability scan as part of your software release pipeline, and do not release if “High Risk” vulnerabilities are identified.



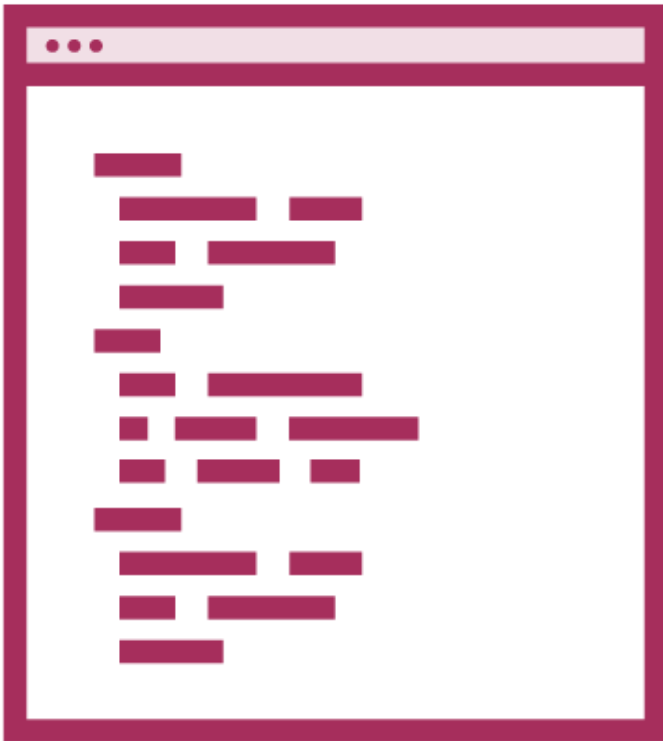
Merge DevOps and Sec to Remove Friction



Security Visibility and Control

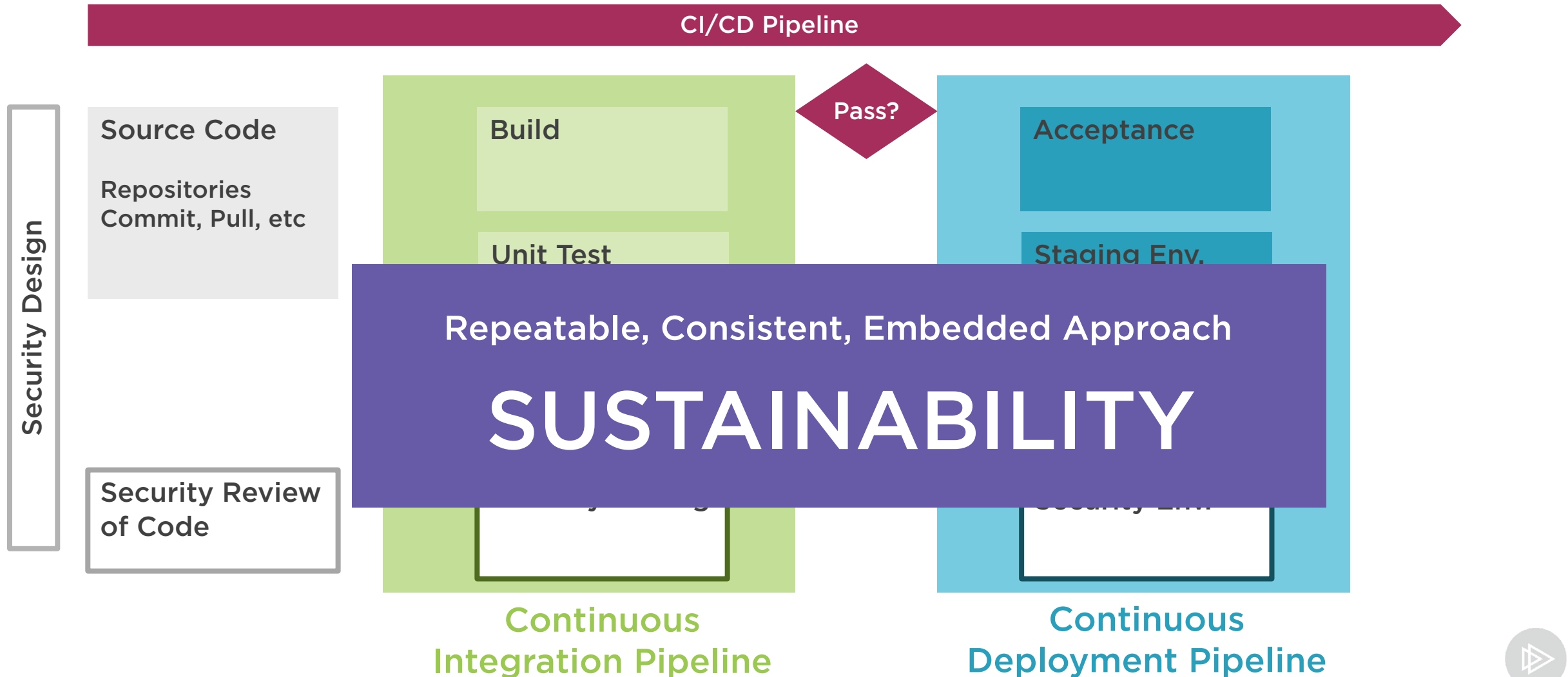


Security as Code



- **Code review** becomes **code preview**
- **Patching** becomes **build new environment and deploy**
- **Incident Response** becomes **Incident Avoidance using Threat Modeling**

Continuous Integration / Delivery (CI/CD)



Summary



Traditional security checks are too slow

DevOps already the solution, add security

Think of security as code or script

Up Next

- Identifying the benefits of DevSecOps

