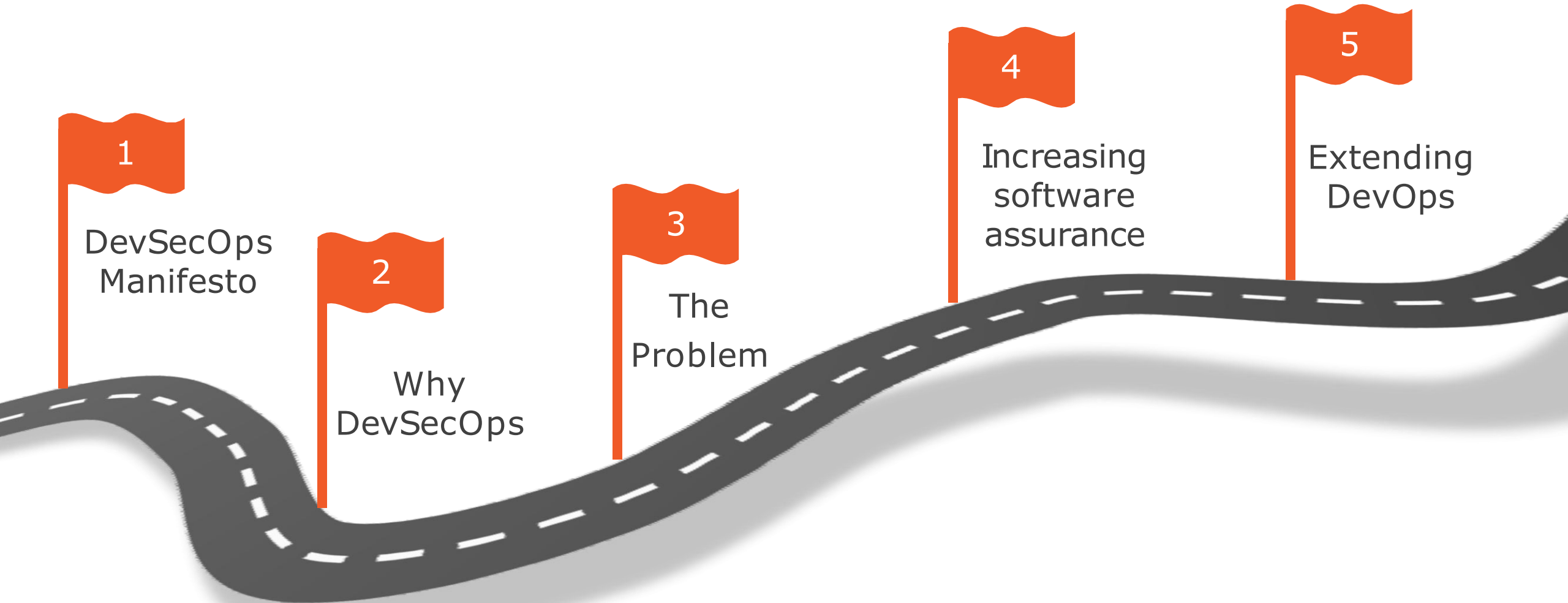


DevSecOps: The Big Picture

UNDERSTANDING DEVSECOPS CONCEPTS

Welcome to Our DevSecOps Journey





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

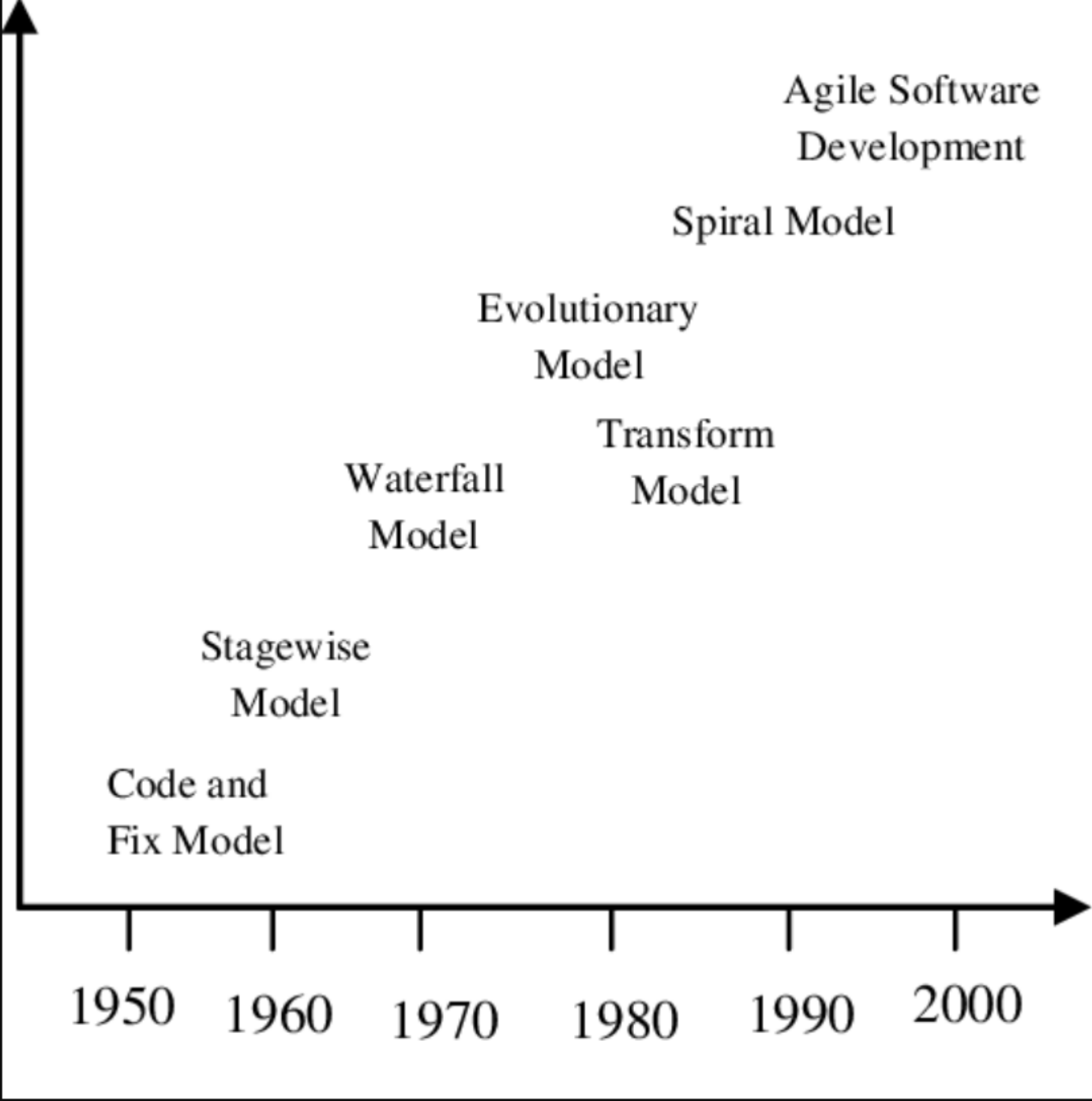
Red & Blue Team Exploit Testing over Relying on Scans & Theoretical Vulnerabilities

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



Software Development Evolution

SOFTWARE DEVELOPMENT EVOLUTION



Waterfall



Agile



DevOps



What's next

Information Security Friction



“Waterfall Methodology”

- 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

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

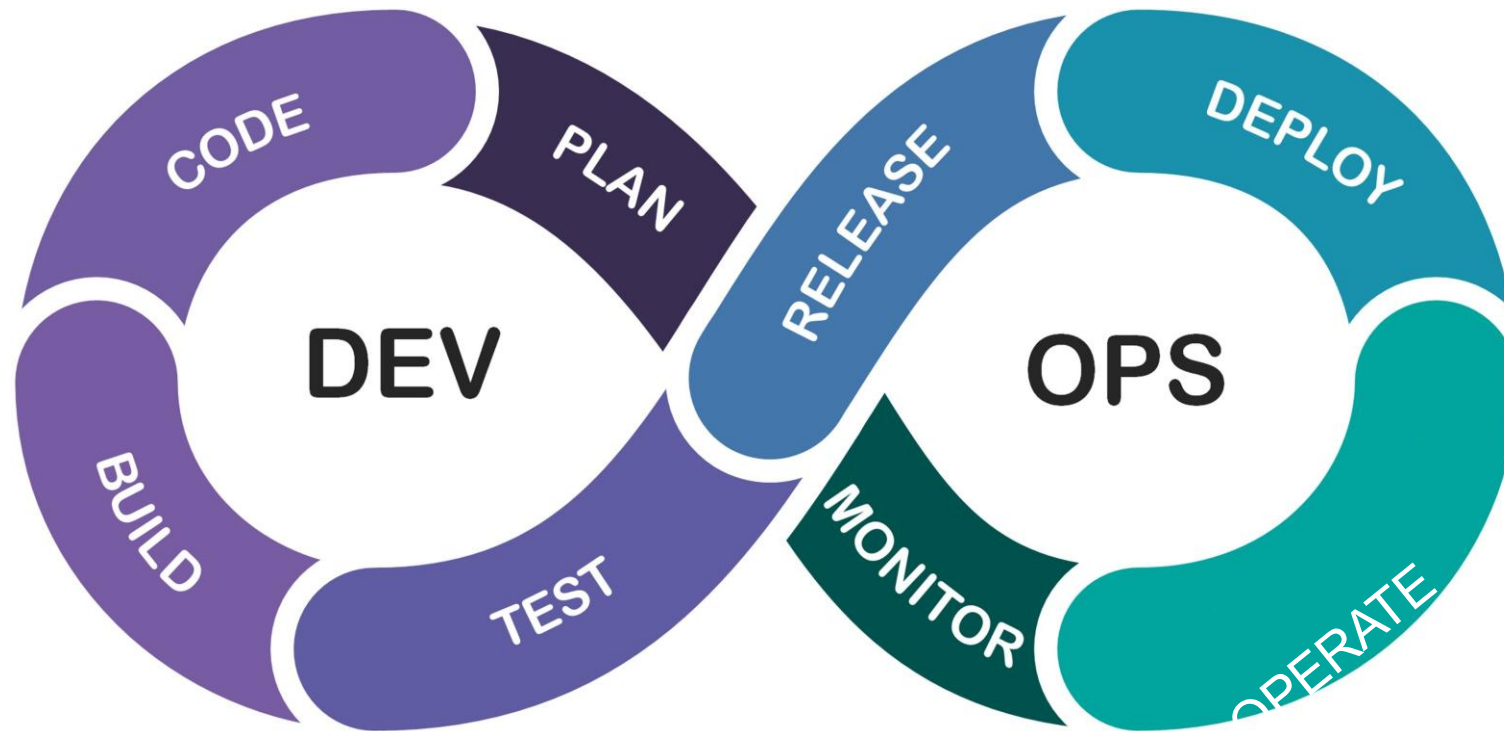
Engineers follow procedures (SOPs)

Rigid Change Advisory Board

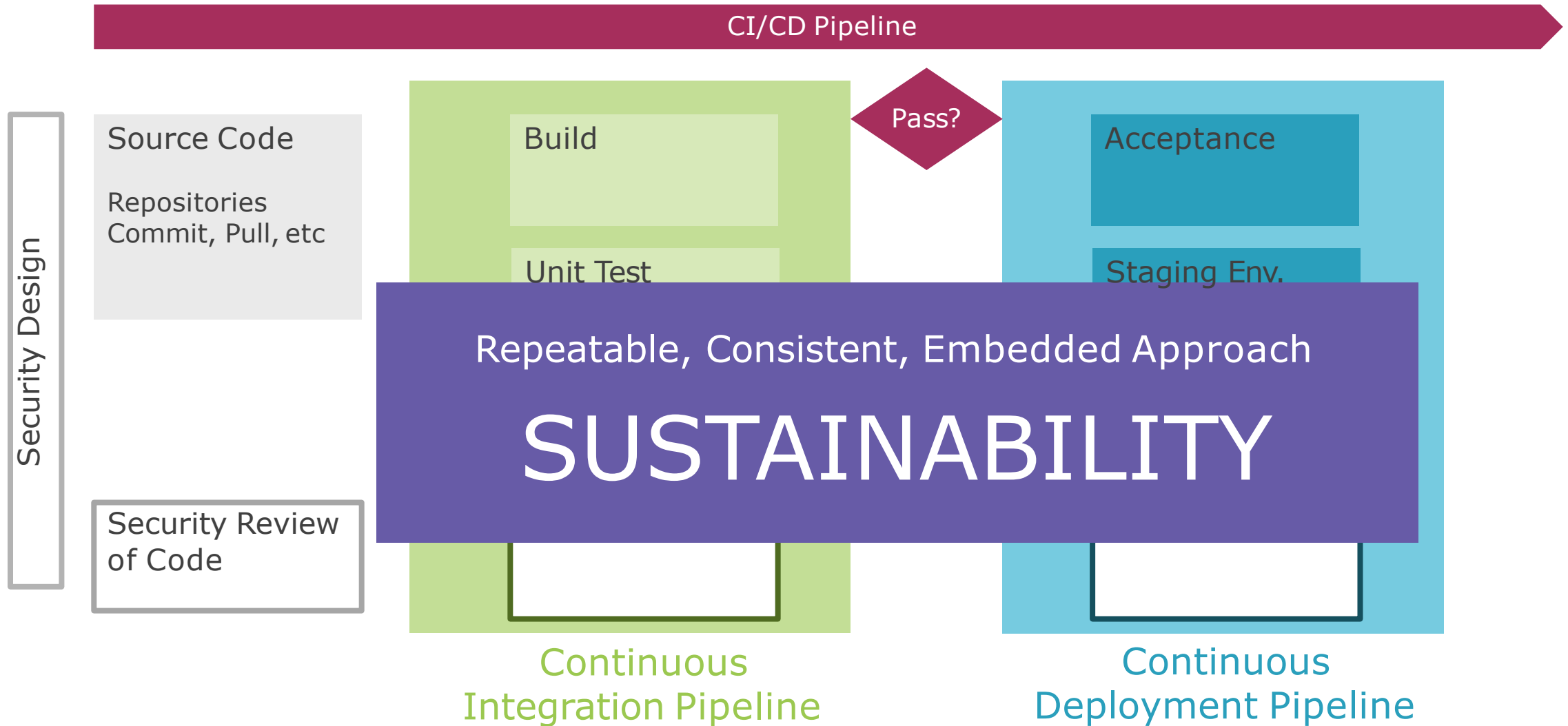
After release, have constant monitoring,
and 24x7 response support



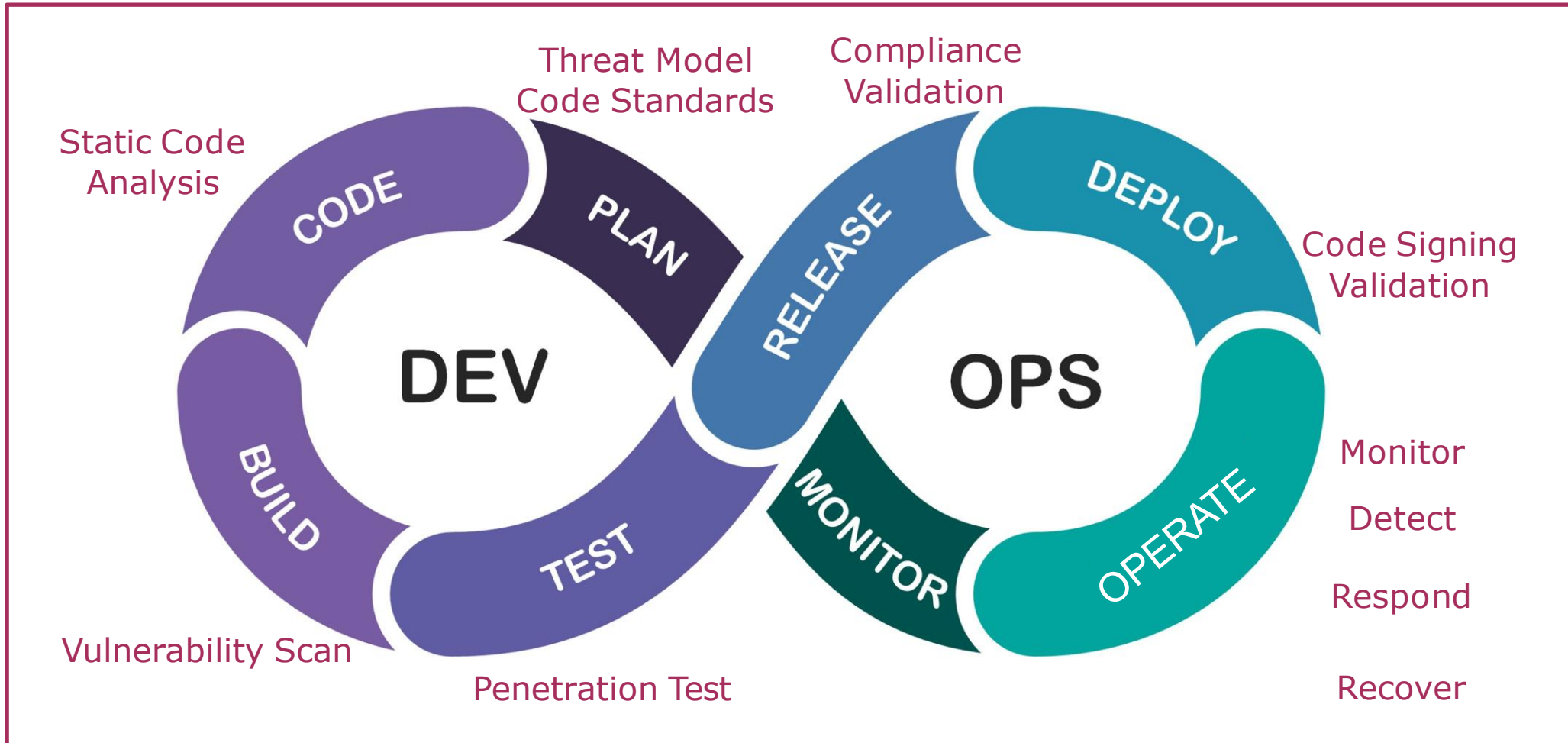
Merge Dev and Ops to Remove Friction



Continuous Integration / Delivery (CI/CD)

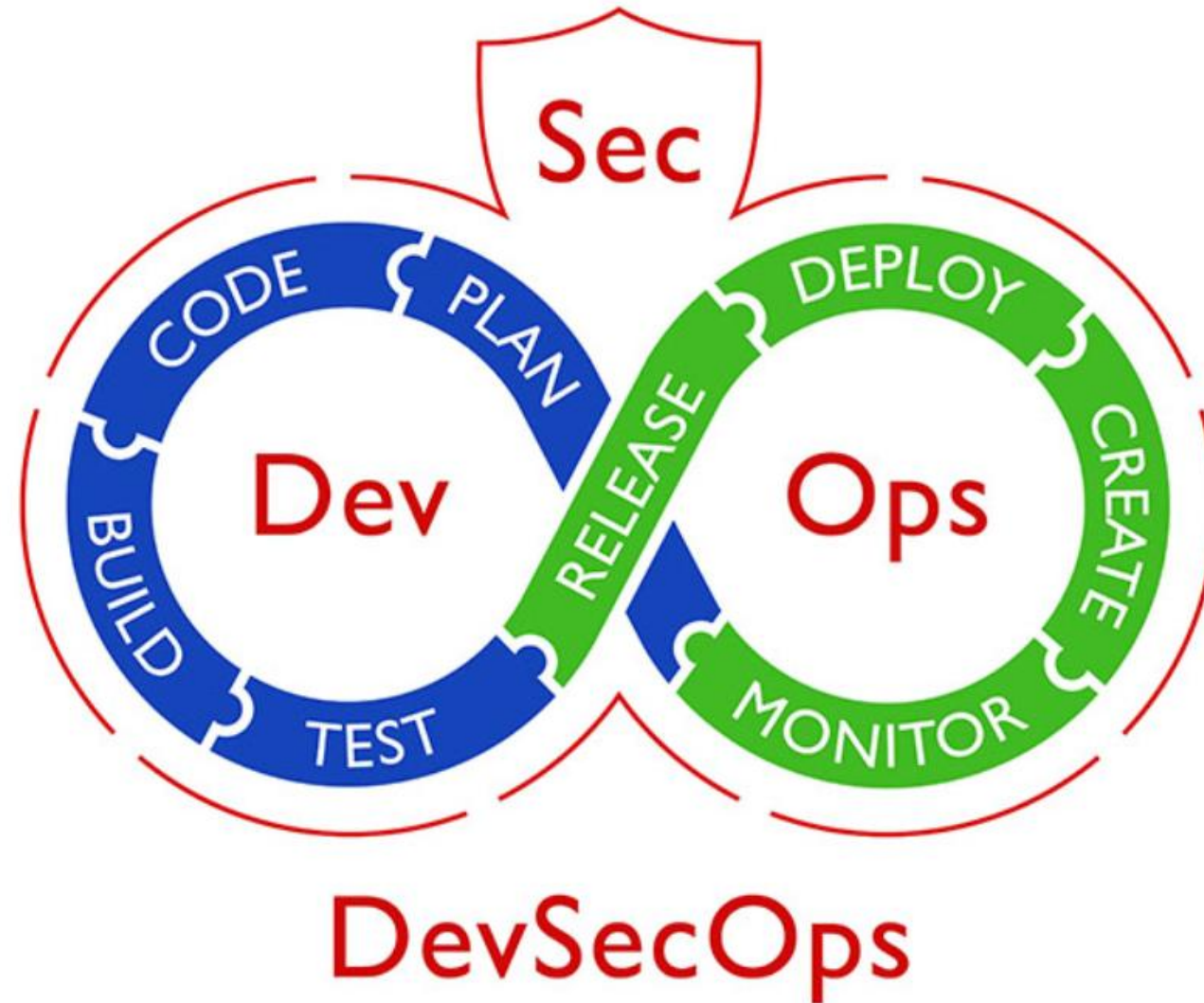


Merge DevOps and Sec to Remove Friction

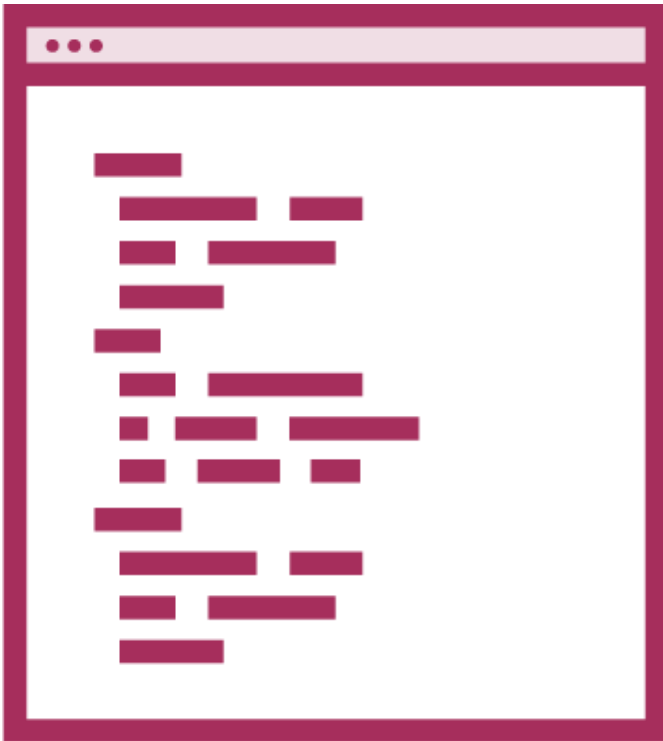


Security Visibility and Control

Merge DevOps and Sec to Remove Friction



Security as Code



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

DevSecOps

Plan

(PRE-PRODUCTION)

Threat modeling, change impact analysis

Deploy

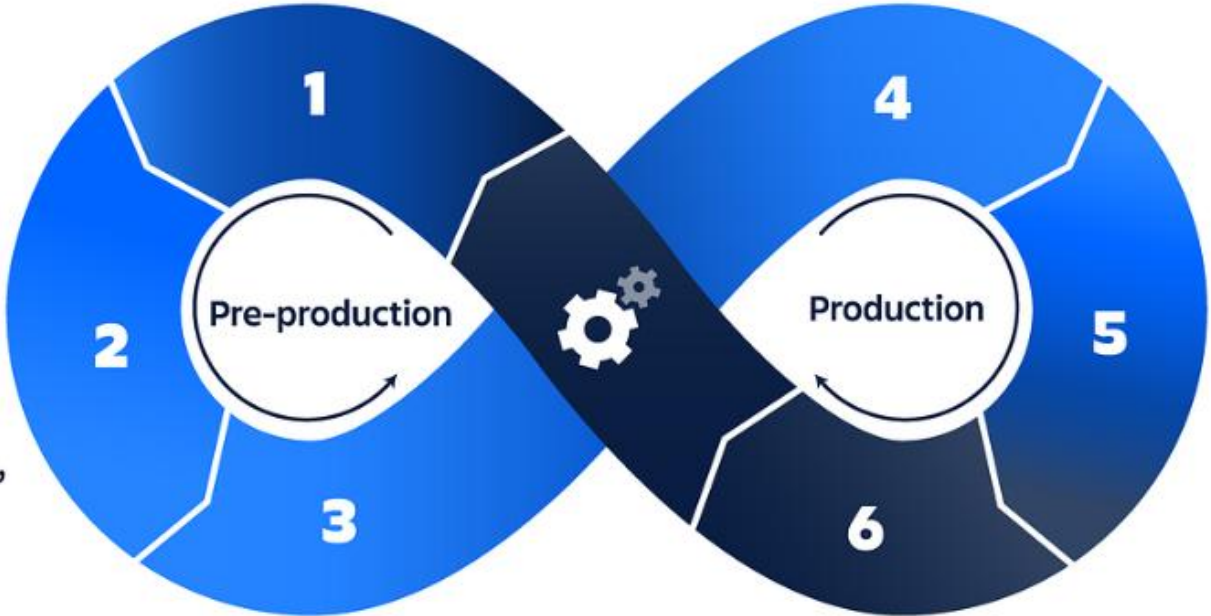
(PRODUCTION)

Access and configuration management, chaos engineering, pen testing

Build

(PRE-PRODUCTION)

Pre-commit hooks, software composition analysis, SAST, code review, container security, vulnerability scanning, DAST



Test

(PRE-PRODUCTION)

DAST

Monitor

(PRODUCTION)

SIEM, vulnerability monitoring, access control

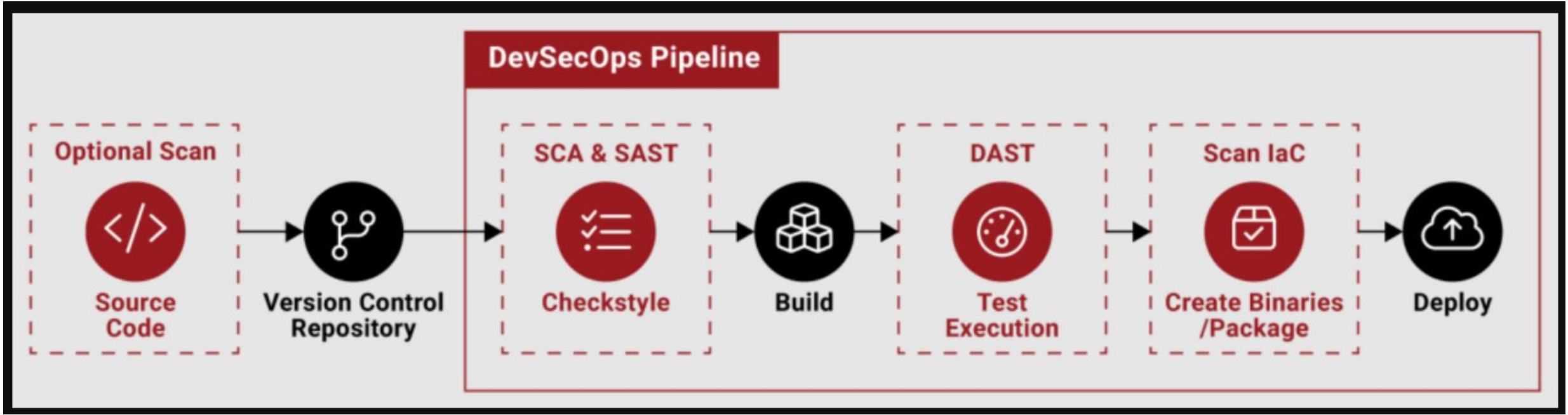
Operate

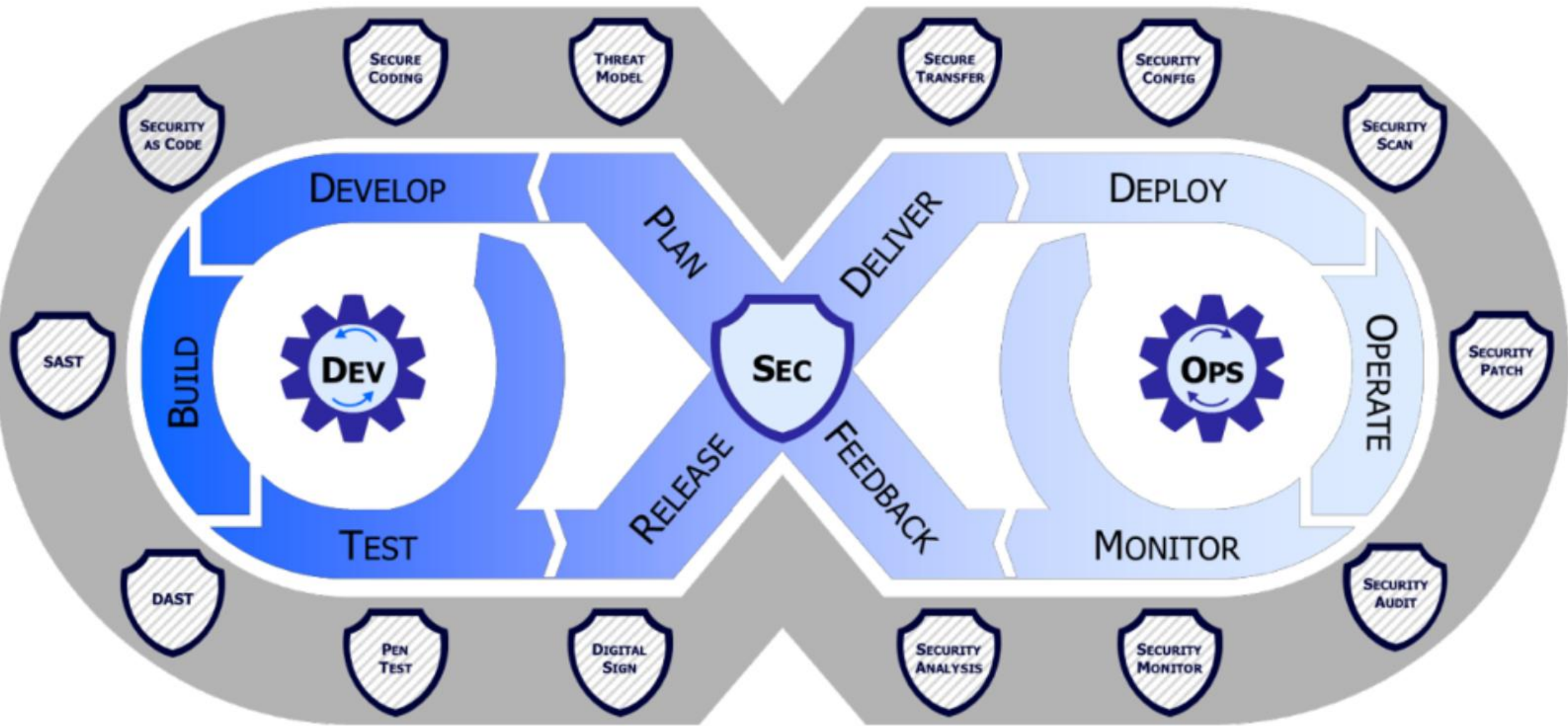
(PRODUCTION)

Log collection, RASP, Patching, WAF

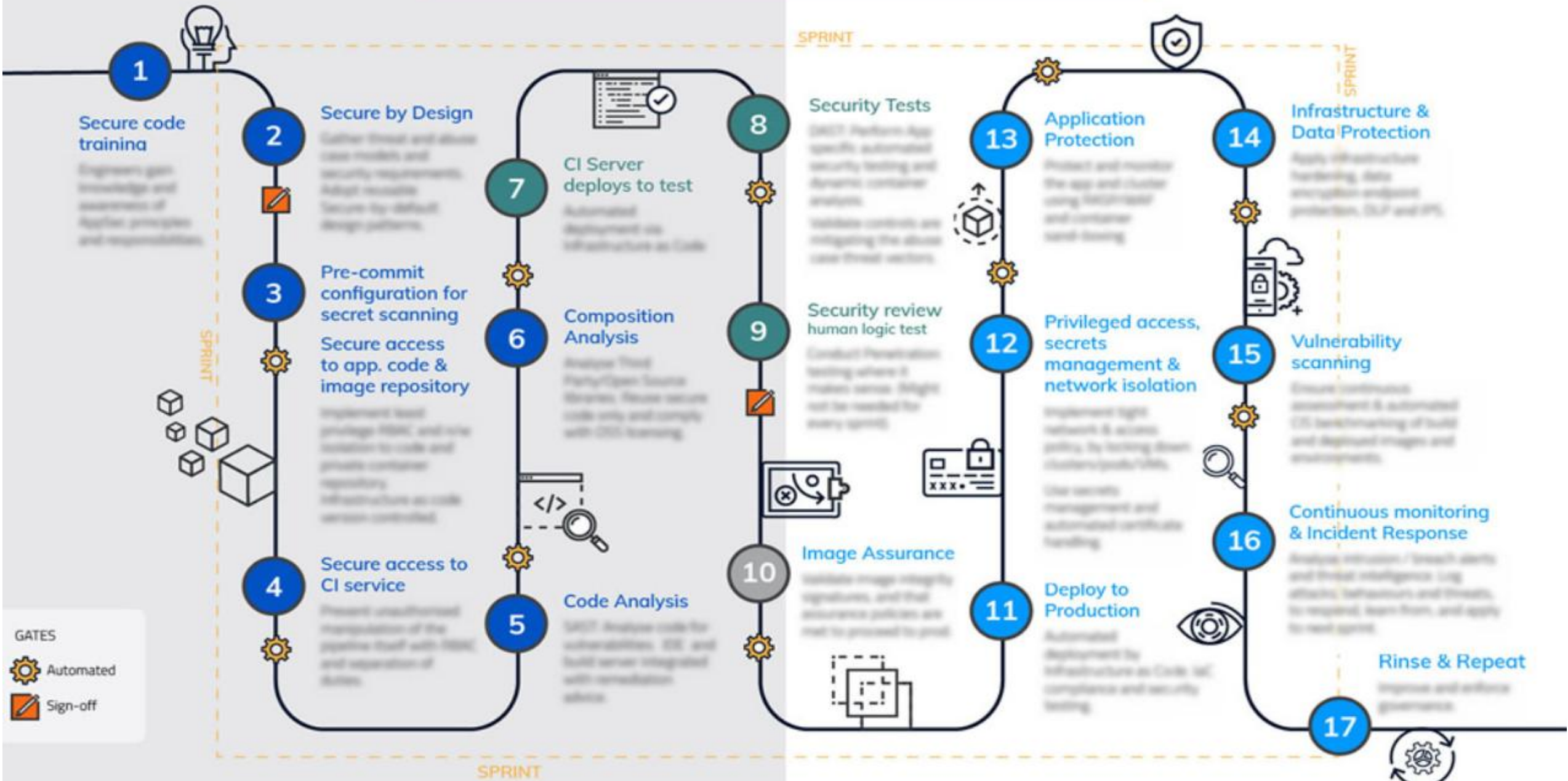
DevSecOps

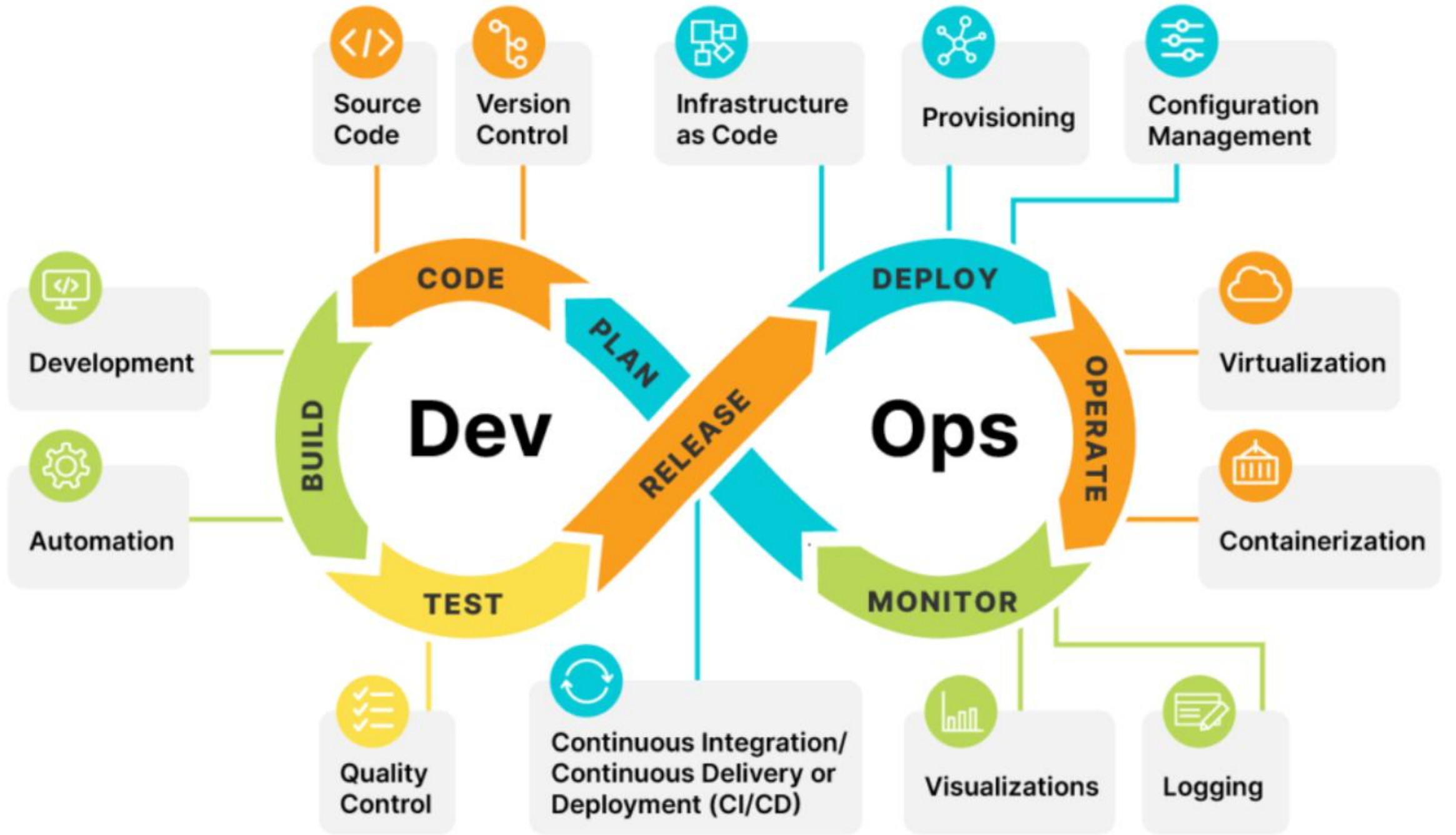




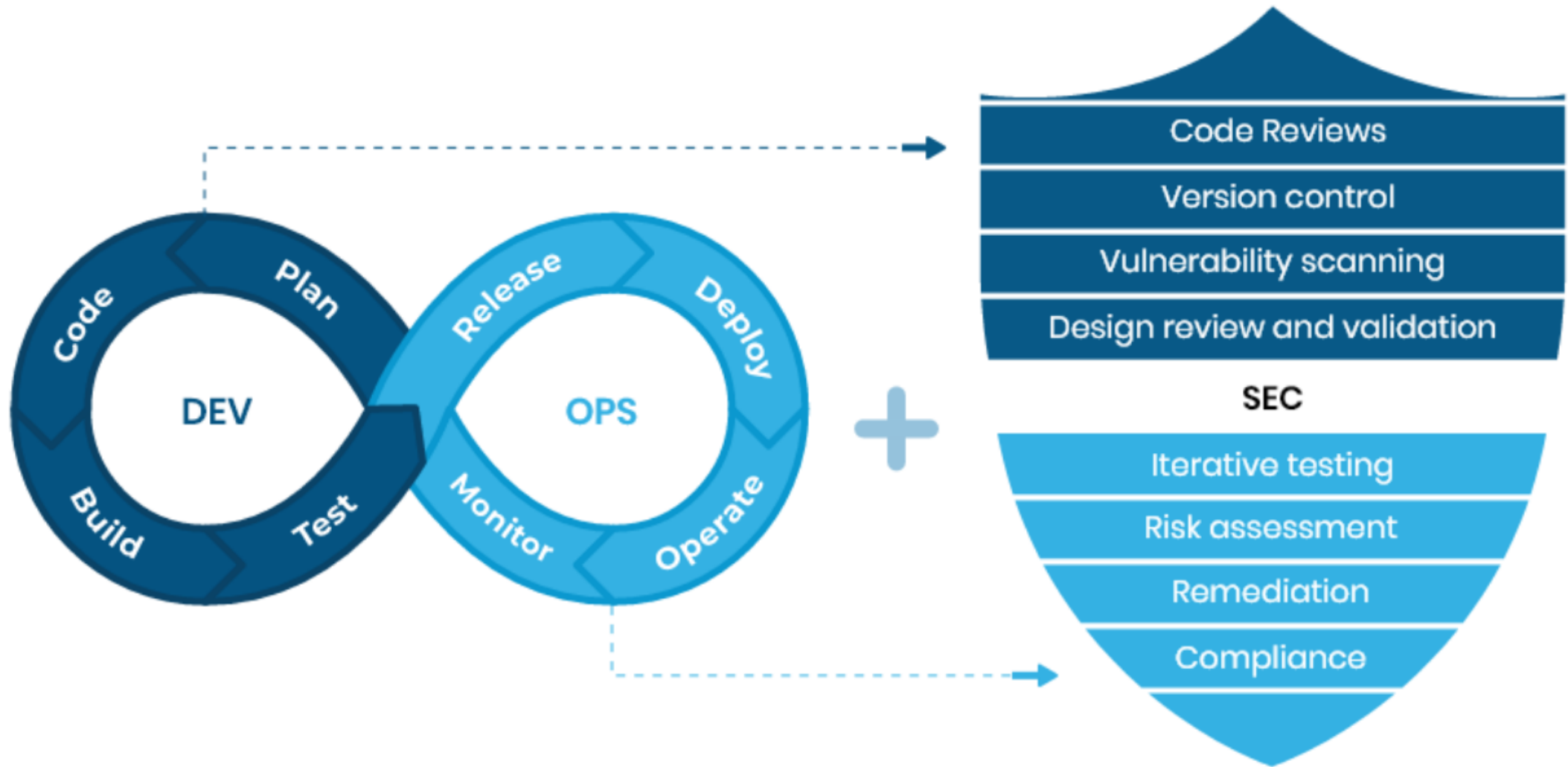


DevSecOps Security Controls





What is DevSecOps?



Identifying the Benefits of DevSecOps

Overview

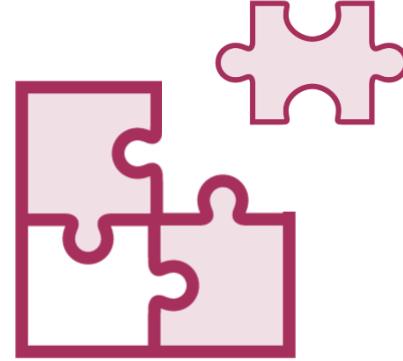
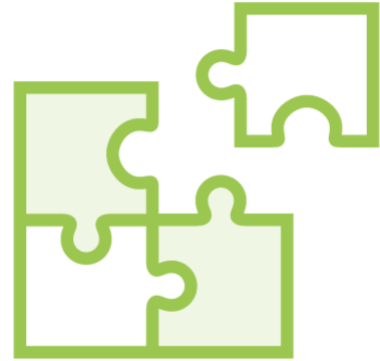


Where is DevSecOps appropriate

Demonstrate benefits of DevSecOps

Roles & responsibilities within DevSecOps

Where is DevSecOps Appropriate?



Where is DevSecOps Appropriate?



Very suitable

Agile methodology

Existing DevOps in place

Many releases per year

Some automation in place

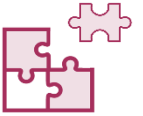
Possibly not as suitable

Waterfall methodology

Highly regulated, requires approval

Few releases per year

Zero automation in place

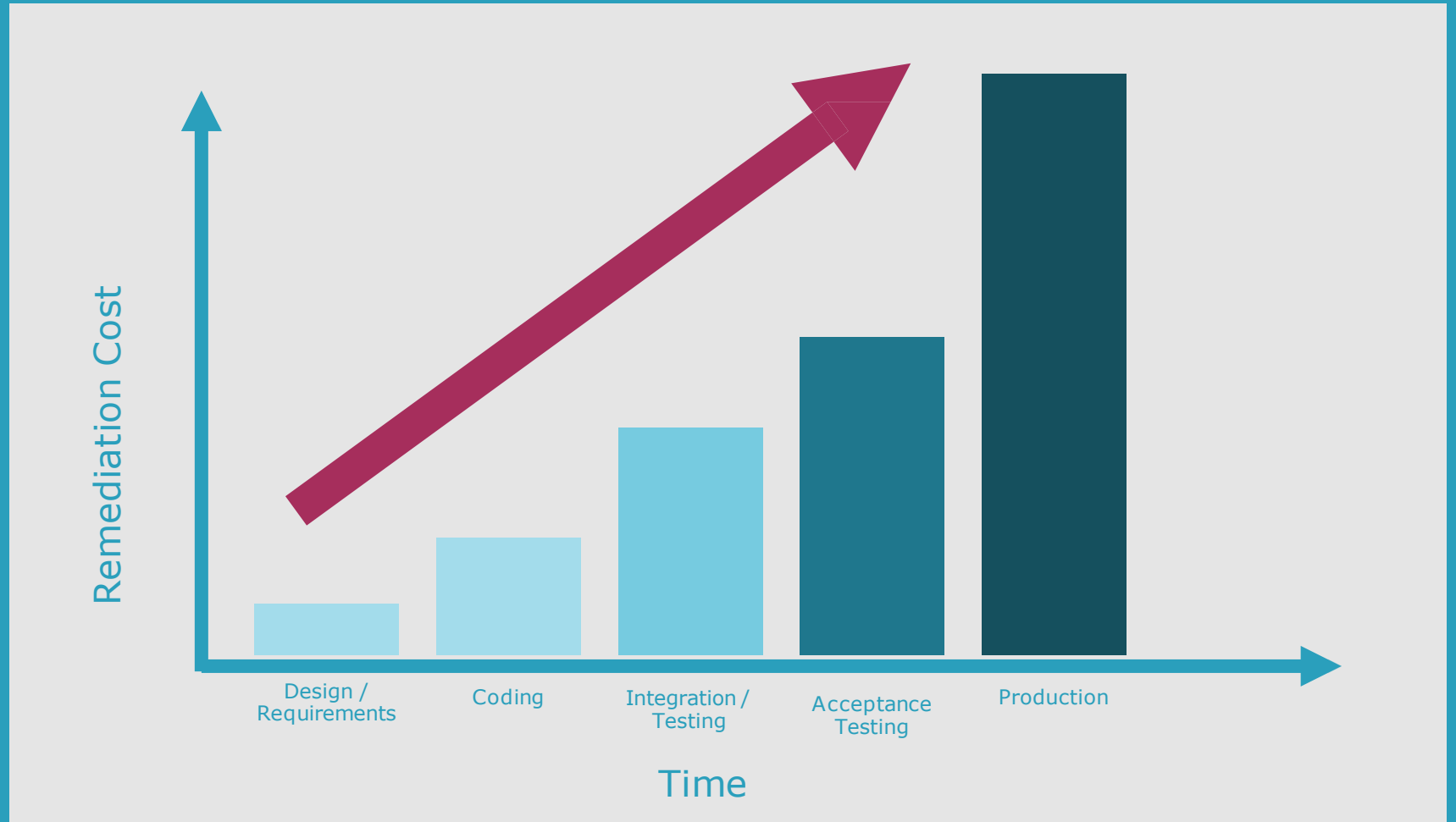


Some Benefits of DevSecOps Practice

- Security issues are found earlier in the lifecycle, reducing the cost of remediation and rework
- Reduce risk for Cotocus and customers
- Increased team collaboration
- Reduce time on rework for security vulnerabilities
- Consistency in approach - continuous security
- Increased compliance levels – ‘Secure by design’
- Better and automated security testing helps compliance with various laws
 - GDPR – General Data Protection Regulation
 - CCPA – California Consumer Privacy Act

Reduce time on rework for security vulnerabilities

- Security issues are found earlier in the lifecycle, reducing the cost of remediation and rework

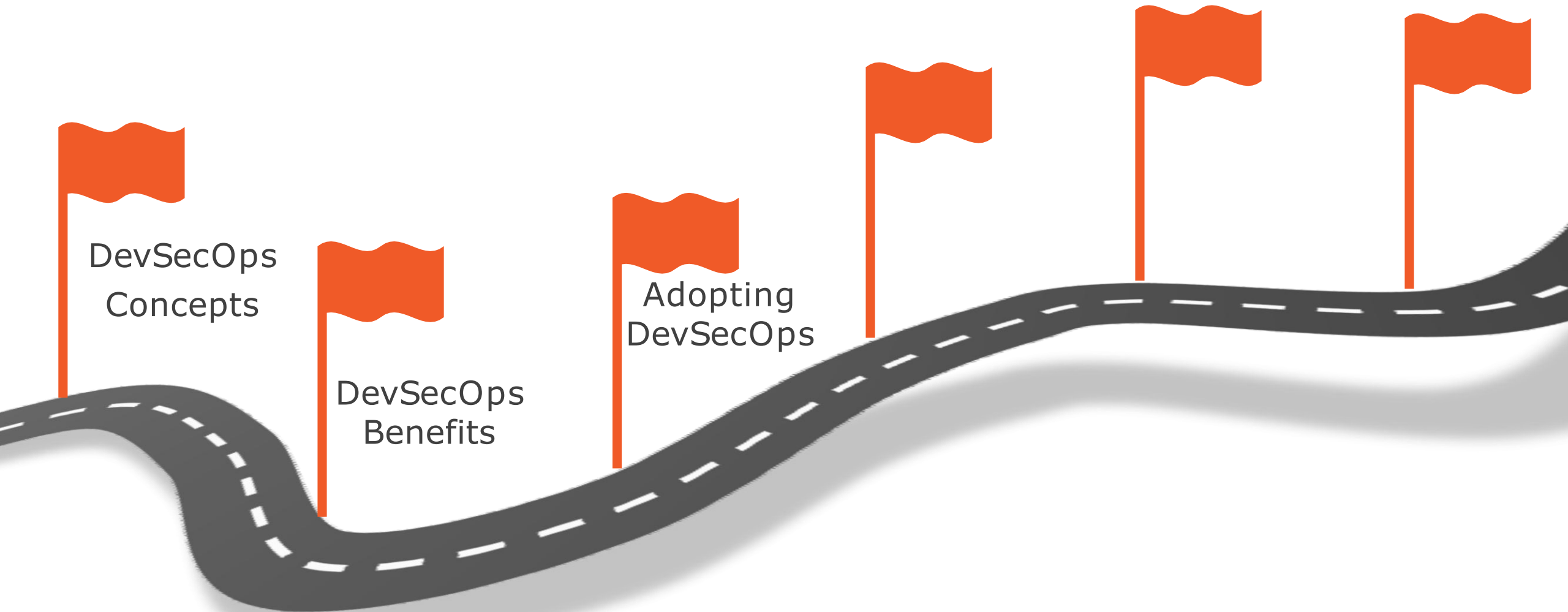


Roles & Responsibilities

Role / Activity	Responsibility	Fulfilled By
Tooling	Ensure the DevOps pipeline is extended to include security testing, integrate with security testing tools.	DevOps Engineer or Software Engineer
Vulnerability Management	Ensure all vulnerabilities are managed, block deployments in accordance with your company's risk policy.	AppSec Engineer or Software Engineer
Application Security	Specialised engineer with focus on security fixes.	AppSec Engineer or Software Engineer
Compliance	Ensure testing meets any compliance or regulatory obligations and evidence is available to demonstrate compliance.	IT Professional or Software Engineer

Adopting DevSecOps in Your Software Development Lifecycle

Continue Our DevSecOps Journey

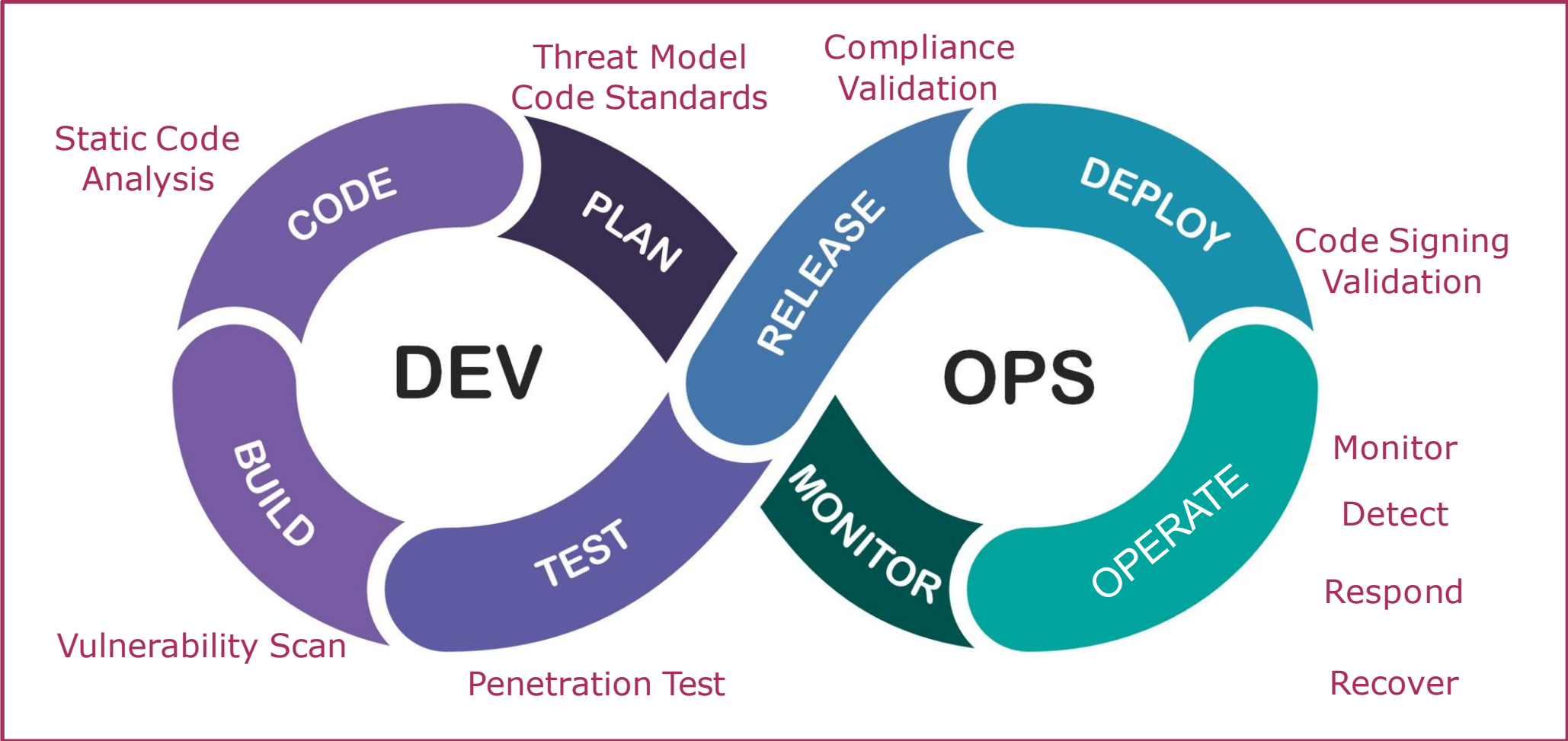


Overview



Positioning DevSecOps in Lifecycle
DevSecOps Maturity Models

Positioning DevSecOps in Your Lifecycle



Security Visibility and Control

DevSecOps Maturity Model

1

	Insanity
Culture	
Skills	
Program / Outcomes	
Security Priorities	

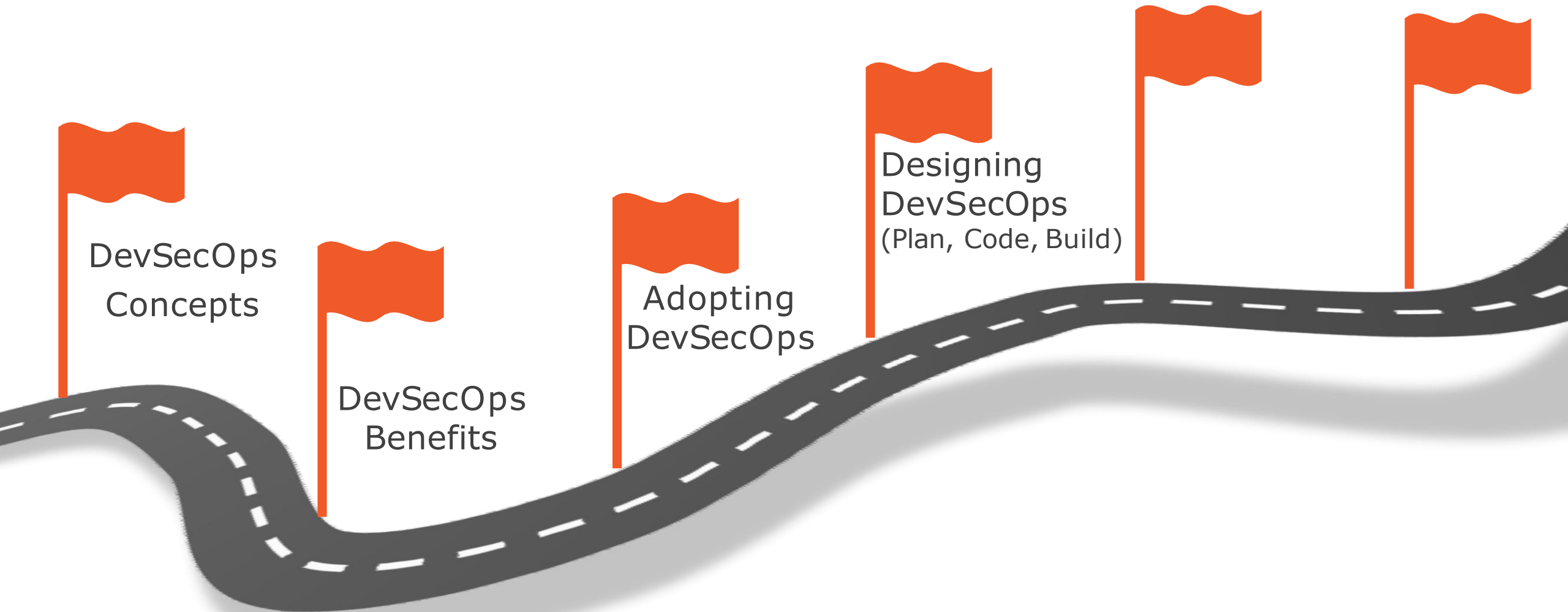
OWASP DevSecOps Maturity Model

1

Dimension	Level 1: Basic understanding of security practices
Build and deployment	
Culture and organization	
Information gathering	
Patch management	
Test and verification	

Designing DevSecOps for Plan, Code, and Build SDLC phases

Continue Our DevSecOps Journey



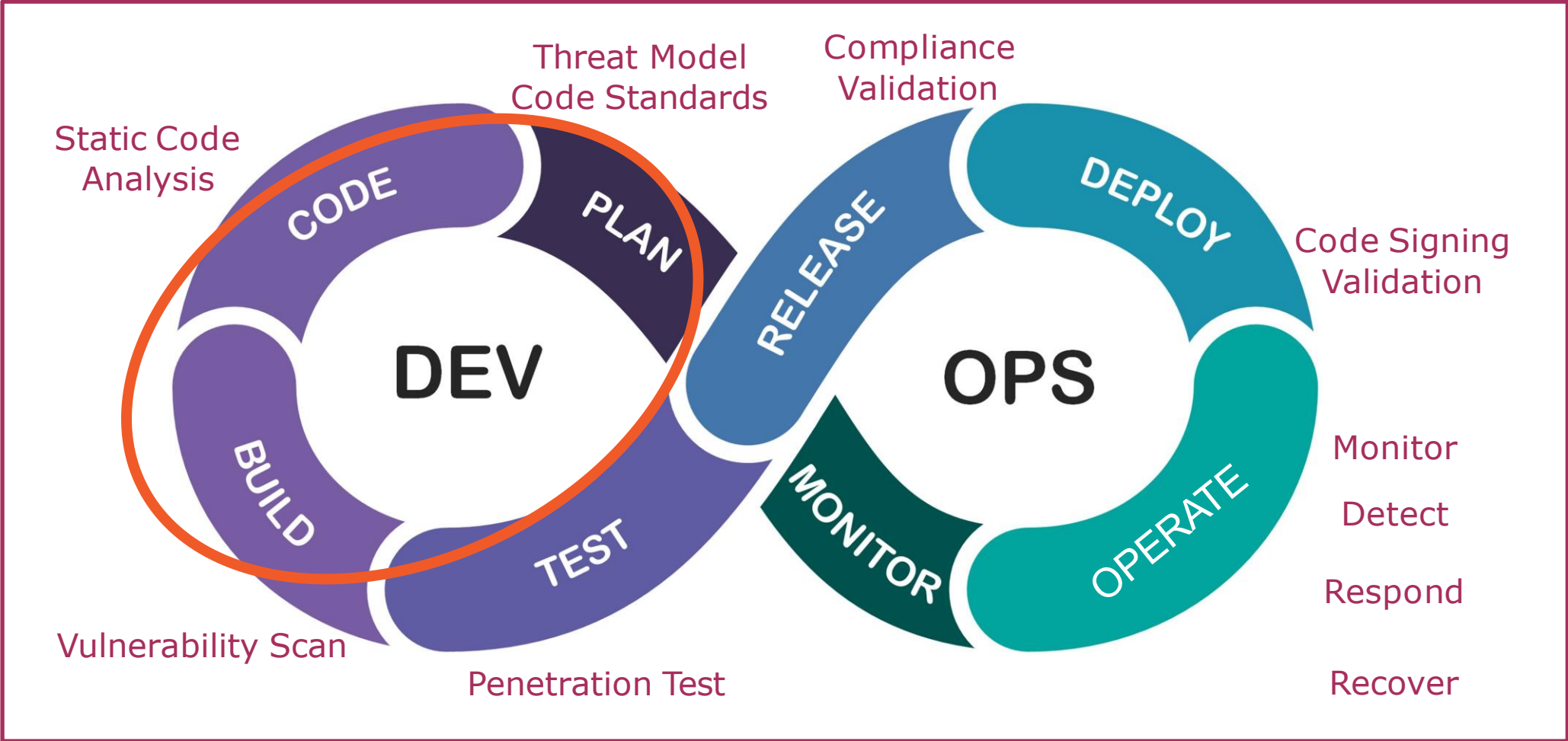
Overview



DevSecOps Requirements for:

- Plan
- Code
- Build

Positioning DevSecOps in Your Lifecycle



Security Visibility and Control

Threat Model

“Threat modeling is a process by which potential threats, such as structural vulnerabilities, can be identified, enumerated, and prioritized – all from a hypothetical attacker’s point of view”

Threat Modelling



- S. Spoofing
- T. Tampering
- R. Repudiation
- I. Information disclosure / leakage
- D. Denial of service
- E. Elevation of privilege

Threat Modelling



- S. Spoofing
- T. Tampering
- R. Repudiation
- I. Information disclosure / leakage
- D. Denial of service
- E. Elevation of privilege



Microsoft Threat Modelling Tool

<https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool>

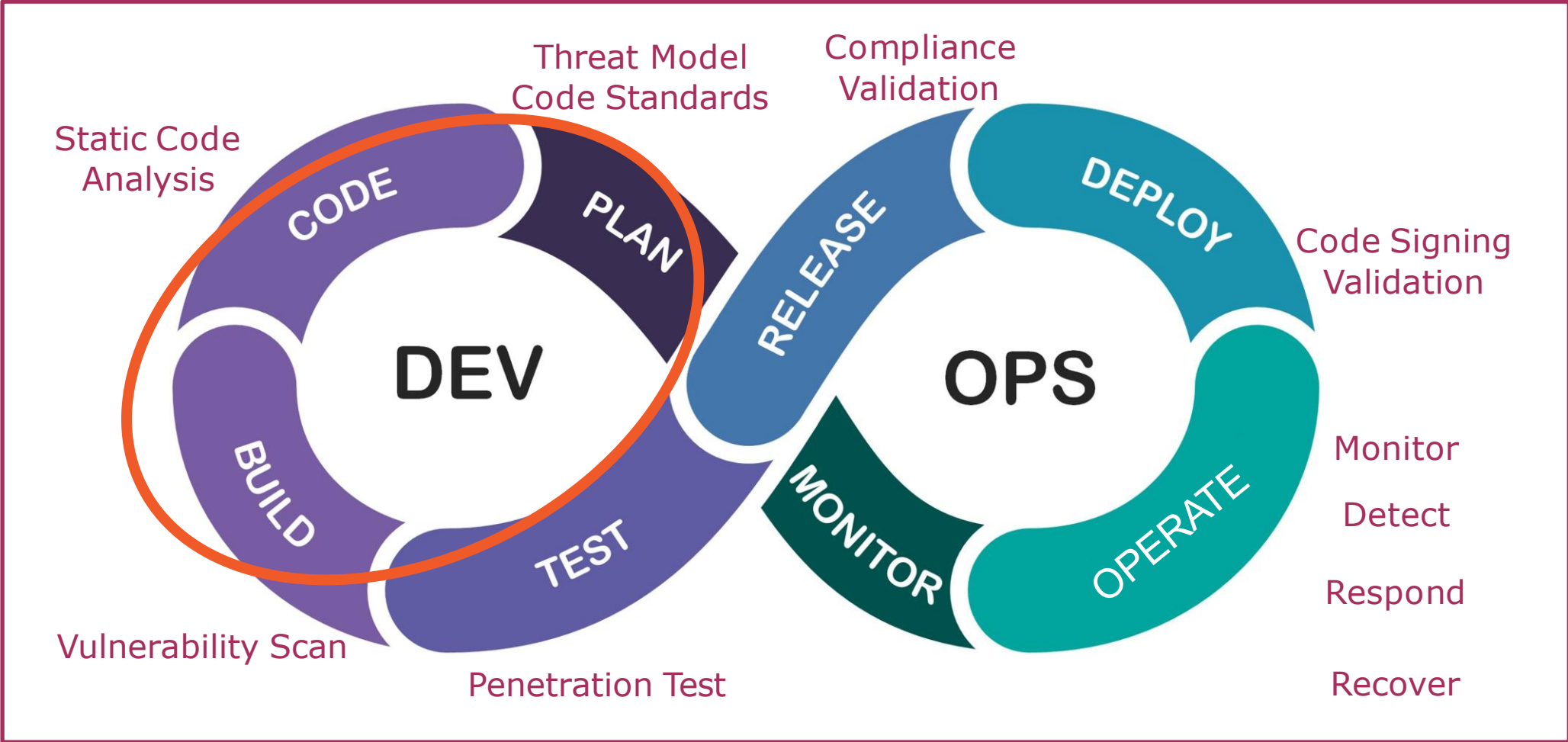
Secure Code Standards



CMU SEI -Top 10 Secure Coding Practice

1. Validate input
2. Heed compiler warnings
3. Architect and design for security
4. Keep it simple
5. Default deny
6. Adhere to principle of least privilege
7. Sanitize data from other systems
8. Practice defense in depth
9. Practice effective quality assurance
10. Adopt a secure coding standard

Positioning DevSecOps in Your Lifecycle



Security Visibility and Control

SAST or SCA

Static Application Security Testing (SAST)

Examines source code to identify weaknesses that can lead to security vulnerabilities

Software Composition Analysis (SCA)

Checks Open Source components against known vulnerabilities

Secure Code Analysis



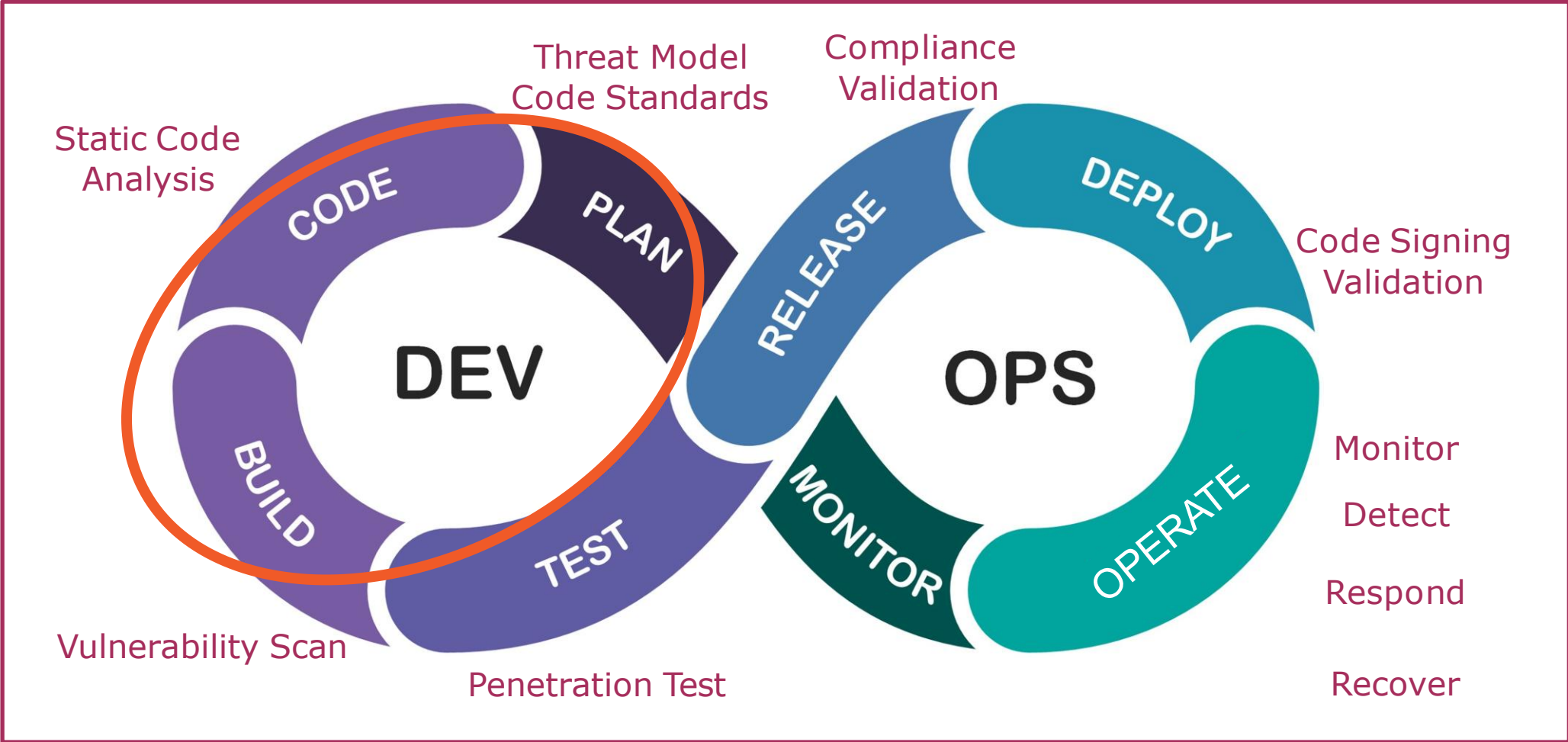
Features of SAST

1. Reads source code
2. Language specific scanner
3. False positives
4. Fast and automated
5. Finds weaknesses early

NIST list of source code security analyzers

https://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html

Positioning DevSecOps in Your Lifecycle



Security Visibility and Control

Vulnerability Scanning

Software Composition Analysis (SCA)

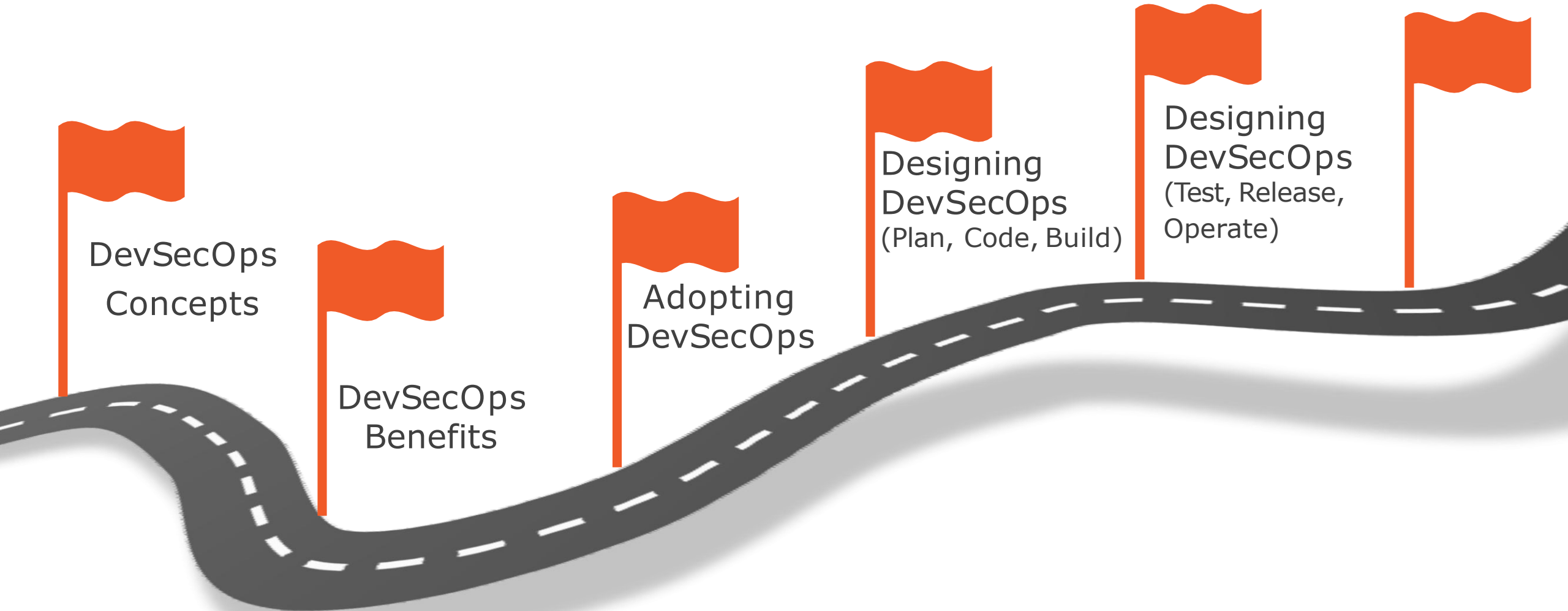
Checks Open Source components against known vulnerabilities

Dynamic Application Security Testing (DAST)

Vulnerability scanners run on completed (compiled) code

Designing DevSecOps for Test, Release, and Operate SDLCphases

Continue Our DevSecOps Journey



DevSecOps
Concepts

DevSecOps
Benefits

Adopting
DevSecOps

Designing
DevSecOps
(Plan, Code, Build)

Designing
DevSecOps
(Test, Release,
Operate)

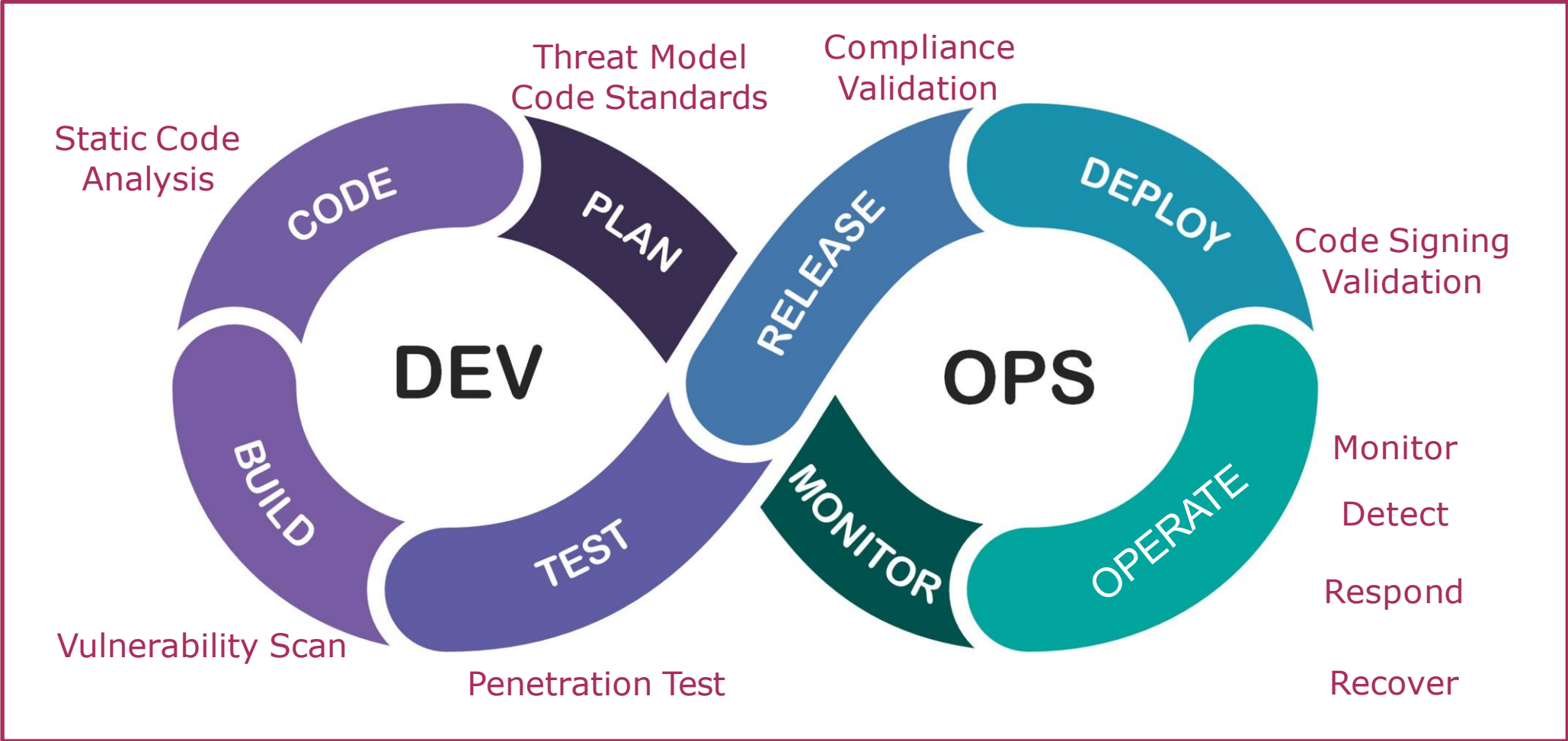
Overview



DevSecOps Requirements for:

- Test
- Release
- Operate

Positioning DevSecOps in Your Lifecycle



Security Visibility and Control

Test Phase

Penetration Testing
(Manual)

Load Testing (DDoS) –putting
demand on system and
measuring its response

Fuzzing

Integration Testing –testing of
combined individual modules

Fuzzing

“Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. The program is then monitored for exceptions such as crashes, failing built-in code assertions, or potential memory leaks. Typically, fuzzers are used to test programs that take structured inputs.”

Deploy Phase

SSL Testing


Ensure all Transport Security
Layer certificates are valid

Application Hardening

Reducing the attack surface
area

SSL Testing

https://www.ssllabs.com/sslttest/analyze.html?d=richardharpur.com&hideResults=on

 Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > richardharpur.com

SSL Report: richardharpur.com

[Scan Another >>](#)

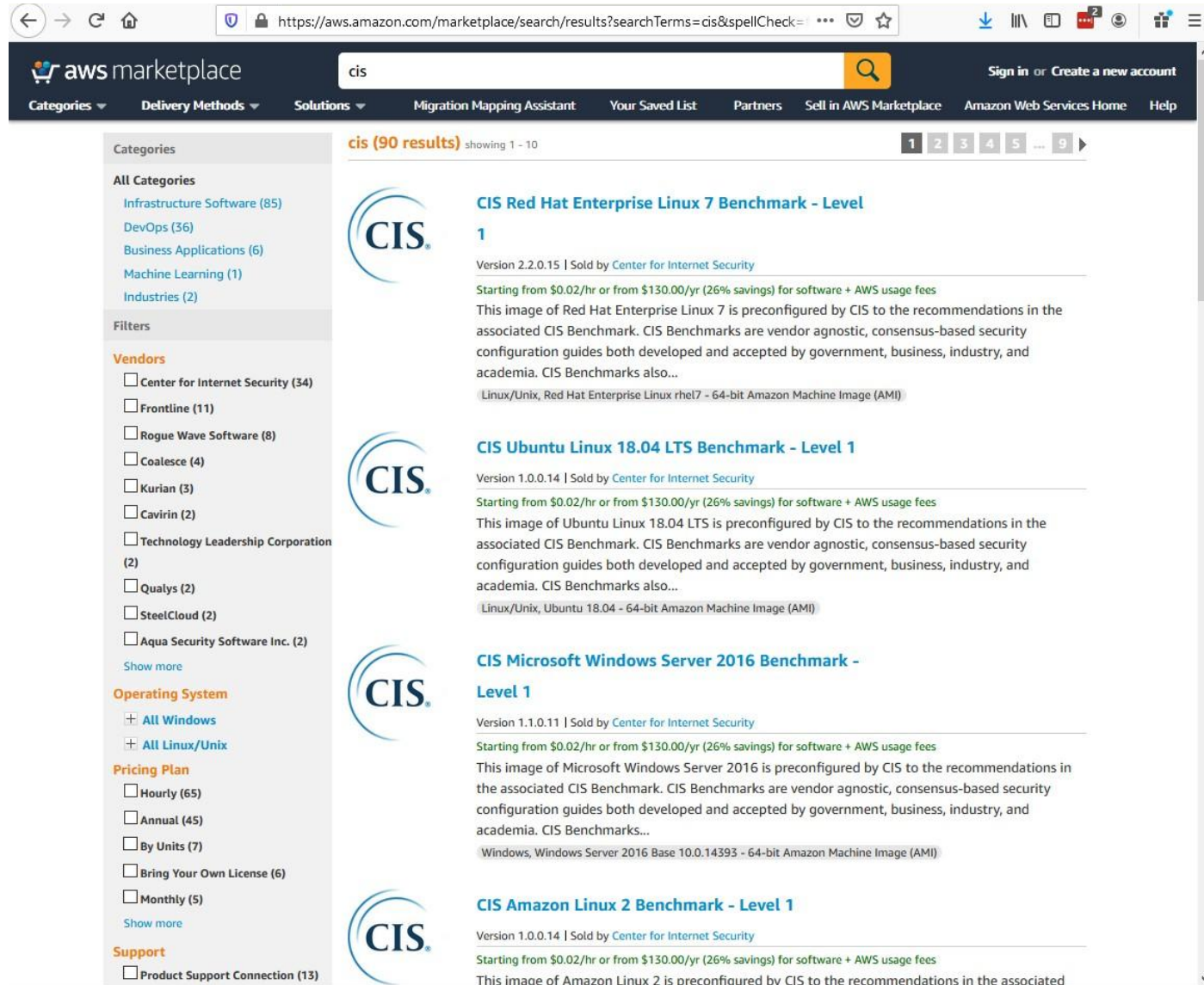
	Server	Test time	Grade
1	2606:4700:3036:0:0:0:6818:7307 Ready	Wed, 20 May 2020 00:22:34 UTC Duration: 45.706 sec	A+
2	2606:4700:3035:0:0:0:6818:7207 Ready	Wed, 20 May 2020 00:23:19 UTC Duration: 43.362 sec	A+
3	104.24.115.7 Ready	Wed, 20 May 2020 00:24:03 UTC Duration: 46.161 sec	A+
4	104.24.114.7 Ready	Wed, 20 May 2020 00:24:49 UTC Duration: 45.73 sec	A+

SSL Report v2.1.4

Copyright © 2009-2020 [Qualys, Inc.](#) All Rights Reserved. [Terms and Conditions](#)

[Try Qualys for free!](#) Experience the award-winning [Qualys Cloud Platform](#) and the entire collection of [Qualys Cloud Apps](#), including [certificate security](#) solutions.

Run Security Hardened Images



The screenshot shows the AWS Marketplace search results for 'cis'. The search bar at the top contains 'cis' and shows 90 results. The left sidebar contains filters for Categories, Vendors, Operating System, Pricing Plan, and Support. The main content area displays four search results, each with a CIS logo, a title, version information, pricing, and a description.

Categories

- All Categories
 - Infrastructure Software (85)
 - DevOps (36)
 - Business Applications (6)
 - Machine Learning (1)
 - Industries (2)

Filters

Vendors

- Center for Internet Security (34)
- Frontline (11)
- Rogue Wave Software (8)
- Coalesce (4)
- Kurian (3)
- Cavirin (2)
- Technology Leadership Corporation (2)
- Qualys (2)
- SteelCloud (2)
- Aqua Security Software Inc. (2)
- [Show more](#)

Operating System

- All Windows
- All Linux/Unix

Pricing Plan

- Hourly (65)
- Annual (45)
- By Units (7)
- Bring Your Own License (6)
- Monthly (5)
- [Show more](#)

Support

- Product Support Connection (13)

Search Results:

cis (90 results) showing 1 - 10

CIS Red Hat Enterprise Linux 7 Benchmark - Level 1
Version 2.2.0.15 | Sold by Center for Internet Security
Starting from \$0.02/hr or from \$130.00/yr (26% savings) for software + AWS usage fees
This image of Red Hat Enterprise Linux 7 is preconfigured by CIS to the recommendations in the associated CIS Benchmark. CIS Benchmarks are vendor agnostic, consensus-based security configuration guides both developed and accepted by government, business, industry, and academia. CIS Benchmarks also...
Linux/Unix, Red Hat Enterprise Linux rhel7 - 64-bit Amazon Machine Image (AMI)

CIS Ubuntu Linux 18.04 LTS Benchmark - Level 1
Version 1.0.0.14 | Sold by Center for Internet Security
Starting from \$0.02/hr or from \$130.00/yr (26% savings) for software + AWS usage fees
This image of Ubuntu Linux 18.04 LTS is preconfigured by CIS to the recommendations in the associated CIS Benchmark. CIS Benchmarks are vendor agnostic, consensus-based security configuration guides both developed and accepted by government, business, industry, and academia. CIS Benchmarks also...
Linux/Unix, Ubuntu 18.04 - 64-bit Amazon Machine Image (AMI)

CIS Microsoft Windows Server 2016 Benchmark - Level 1
Version 1.1.0.11 | Sold by Center for Internet Security
Starting from \$0.02/hr or from \$130.00/yr (26% savings) for software + AWS usage fees
This image of Microsoft Windows Server 2016 is preconfigured by CIS to the recommendations in the associated CIS Benchmark. CIS Benchmarks are vendor agnostic, consensus-based security configuration guides both developed and accepted by government, business, industry, and academia. CIS Benchmarks...
Windows, Windows Server 2016 Base 10.0.14393 - 64-bit Amazon Machine Image (AMI)

CIS Amazon Linux 2 Benchmark - Level 1
Version 1.0.0.14 | Sold by Center for Internet Security
Starting from \$0.02/hr or from \$130.00/yr (26% savings) for software + AWS usage fees
This image of Amazon Linux 2 is preconfigured by CIS to the recommendations in the associated

Operate Phase

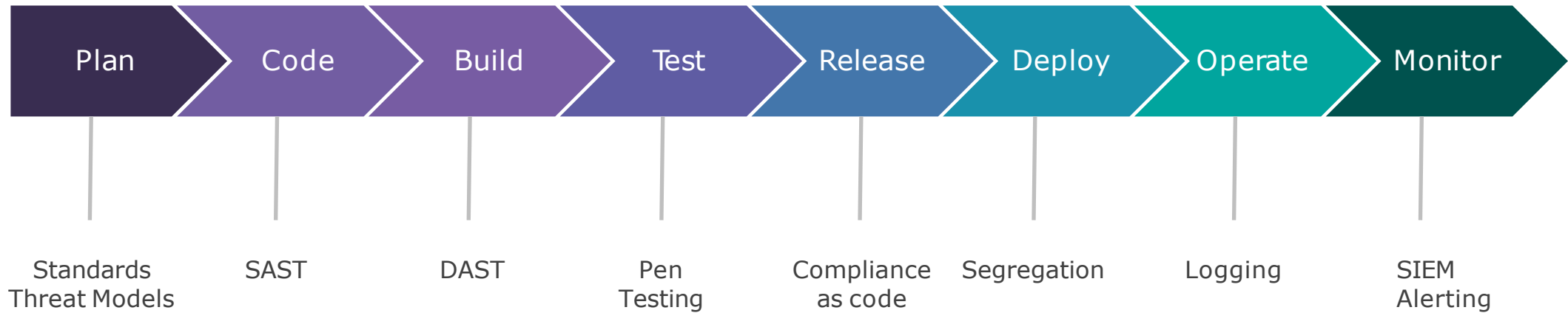
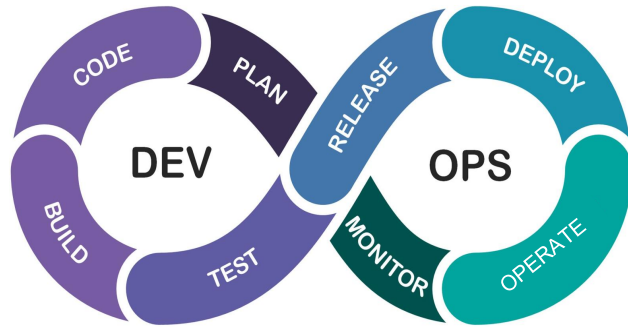
Compliance as code

Check against approved baseline, eliminate any deviation from baseline

Verification and monitoring

Continuous checking that everything is operating "normally"

CI/CD Pipeline



PERIODIC TABLE OF DEVOPS TOOLS (V3)

EMBED

1 Os GI GitLab																2 En Sp Splunk	
3 Fm Gh GitHub	4 En Dt Datacal																
11 Os Sv Subversion	12 En Db DBMaestro																
19 En Cw ISPW	20 En Dp Delphix	21 Os Jn Jenkins	22 Fm Cs Codeship	23 Os Fn FitNesse	24 Fr Ju JUnit	25 Fr Ka Karma	26 Fm Su SoapUI	27 En Ch Chef	28 Fr Tf Terraform	29 En Xld XebiaLabs XL Deploy	30 En Ud UrbanCode Deploy	31 Os Ku Kubernetes	32 Fm Cc CA CD Director	33 En Pr Plutora Release	34 Pd Al Alibaba Cloud	35 Os Os OpenStack	36 Os Ps Prometheus
37 Os At Artifactory	38 En Rg Redgate	39 Pd Ba Bamboo	40 Fm Vs VSTS	41 Fr Se Selenium	42 Fr Jm JMeter	43 Os Ja Jasmine	44 Pd Sl Sauce Labs	45 Os An Ansible	46 Os Ru Rudder	47 En Oc Octopus Deploy	48 Os Go GoCD	49 Os Ms Mesos	50 Pd Gke GKE	51 Fm Om OpenMake	52 Pd Cp AWS CodePipeline	53 Os Cy Cloud Foundry	54 En It ITRS
55 Os Nx Nexus	56 Os Fw Flyway	57 Os Tr Travis CI	58 Fm Tc TeamCity	59 Os Ga Gatling	60 Fr Tn TestNG	61 Fm Tt Tricentis Tosca	62 Pd Pe Perfecto	63 En Pu Puppet	64 Os Pa Packer	65 Fm Cd AWS CodeDeploy	66 En Ec ElectricCloud	67 Os Ra Rancher	68 Pd Aks AKS	69 Os Rk Rkt	70 Os Sp Spinnaker	71 Os Ir iron.io	72 Pd Mg Moogsoft
73 Fm Bb BitBucket	74 En Pf Perforce HelixCore	75 Fm Cr Circle CI	76 Pd Cb AWS CodeBuild	77 Fr Cu Cucumber	78 Os Mc Mocha	79 Os Lo Locust.io	80 En Mf Micro Focus UFT	81 Os Sl Salt	82 Os Ce CFEngine	83 En Eb ElasticBox	84 En Ca CA Automic	85 En De Docker Enterprise	86 Pd Ae AWS ECS	87 Fm Cf Codefresh	88 Os Hm Helm	89 Os Aw Apache OpenWhisk	90 Os Ls Logstash

- Os: Open Source
- Fr: Free
- Fm: Freemium
- Pd: Paid
- En: Enterprise
- Source Control Mgmt.
- Database Automation
- Continuous Integration
- Testing
- Configuration
- Deployment
- Containers
- Release Orchestration
- Cloud
- AIOps
- Analytics
- Monitoring
- Security
- Collaboration

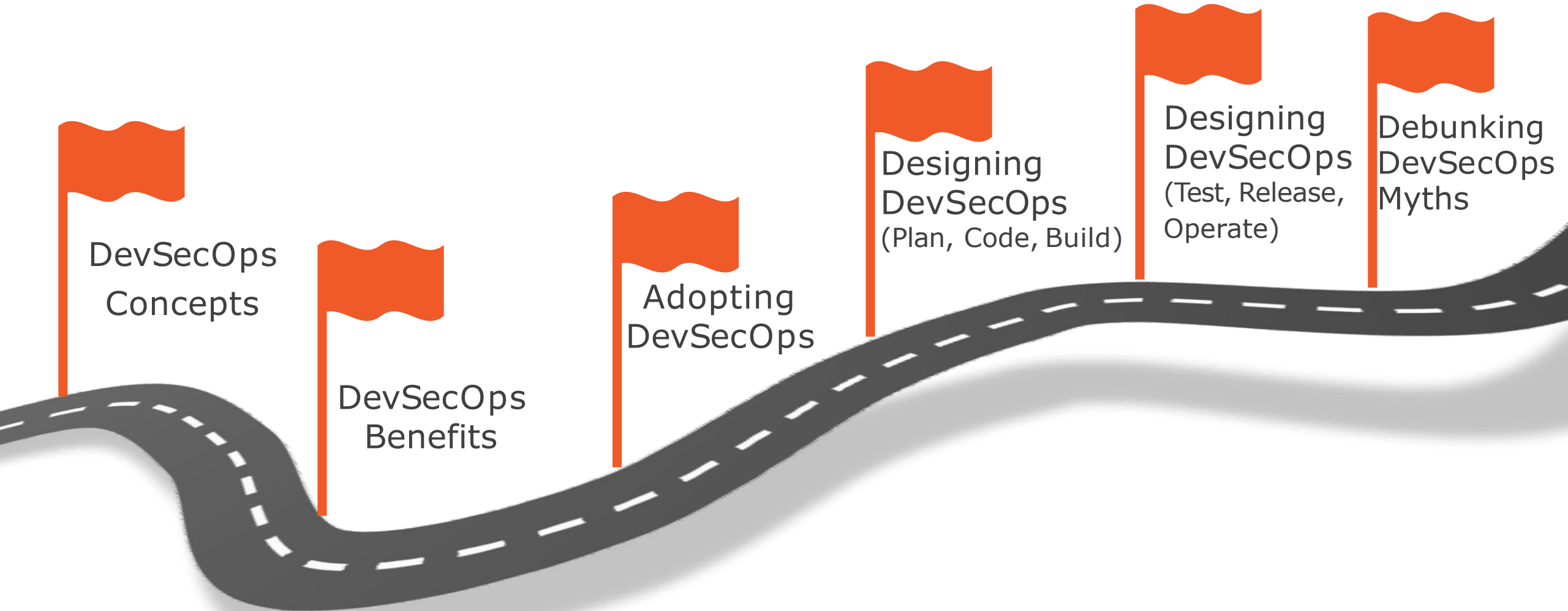
Follow @xebialabs

Publication Guidelines

91 En Xli XebiaLabs XL Impact	92 Os Ki Kibana	93 Fm Nr New Relic	94 En Dt Dynatrace	95 En Dd Datadog	96 Fm Ad AppDynamics	97 Os EI ElasticSearch	98 Os Ni Nagios	99 Os Zb Zabbix	100 En Zn Zenoss	101 En Cx Checkmarx SAST	102 En Sg Signal Sciences	103 En Bd BlackDuck	104 Os Sr SonarQube	105 Os Hv HashiCorp Vault
106 En Sw ServiceNow	107 Pd Jr Jira	108 Fm Tl Trello	109 Fm Sl Slack	110 Fm St Stride	111 En Cn CollabNet VersionOne	112 En Ry Remedy	113 En Ac Agile Central	114 Pd Og OpsGenie	115 Pd Pd Pagerduty	116 Os Sn Snort	117 Os Tw Tripwire	118 En Ck CyberArk Conjur	119 En Vc Veracode	120 En Ff Fortify SCA

Debunking DevSecOps Myths

Continue Our DevSecOps Journey



DevSecOps
Concepts

DevSecOps
Benefits

Adopting
DevSecOps

Designing
DevSecOps
(Plan, Code, Build)

Designing
DevSecOps
(Test, Release,
Operate)

Debunking
DevSecOps
Myths

Overview



Debunking DevSecOps myths

- Special teams
- Delays to deployments
- Buying DevSecOps

DevSecOps Manifesto Reminder

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

We Cannot Introduce DevSecOpsBecause...

We Cannot Introduce DevSecOps Because...

We need a special team dedicated to DevSecOps

“DevSecOps is...
empowered engineering teams taking ownership of how their
product performs all the way to production, including security.”

Larry Maccherone

We Cannot Introduce DevSecOps Because...

The security team still needs to do all security checks for us

By handing over security checks to other teams we introduce delays and time lag into our agile process, instead we should ask the security team to codify their checks so we can build them into our development process automatically.

We Cannot Introduce DevSecOps Because...

We don't have enough resources to do DevSecOps, just buy a tool that does it for us

DevSecOps is not about a capability, it is about a culture, buying a tool is not culture changing. Whilst tools are required to make DevSecOps possible these tools supplement the existing development process and help you deliver DevSecOps, if you have the correct culture in place.

We Cannot Introduce DevSecOps Because...

DevSecOps will just slow down our developers

DevSecOps is about empowering developers to ensure their product gets to production with appropriate security built-in. Traditional approaches to security required testing after developers complete coding and before deployment to production.

Because this is so late in the lifecycle it takes longer to fix and retest software compared to identifying the issue at an earlier stage in the lifecycle, so DevSecOps can save time and increase developer speed.

We Cannot Introduce DevSecOps Because...

DevSecOps will result in our developers giving up control and won't be able to plan

With DevSecOps developers gain control by running security checks at the best possible opportunity to help developers fix the issues quickly and easily. No longer are developers dependent on external teams, and gain control of the work and schedule.