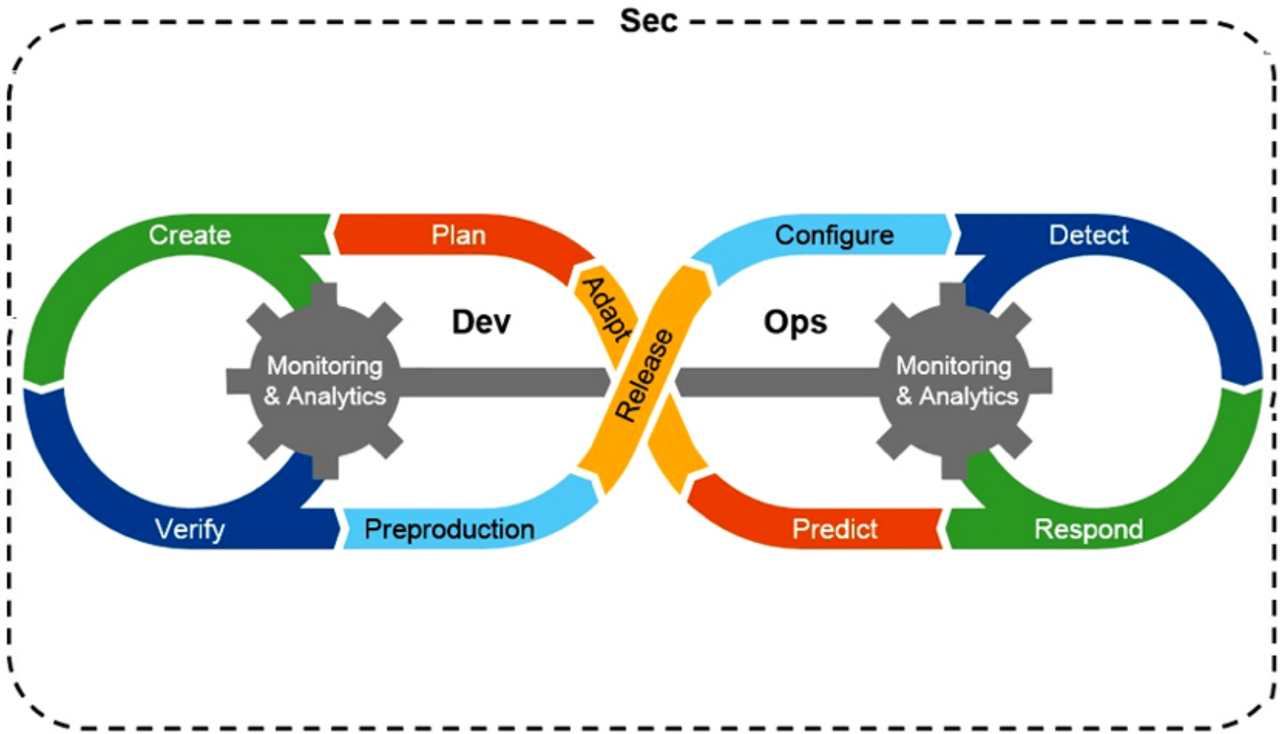
Outline

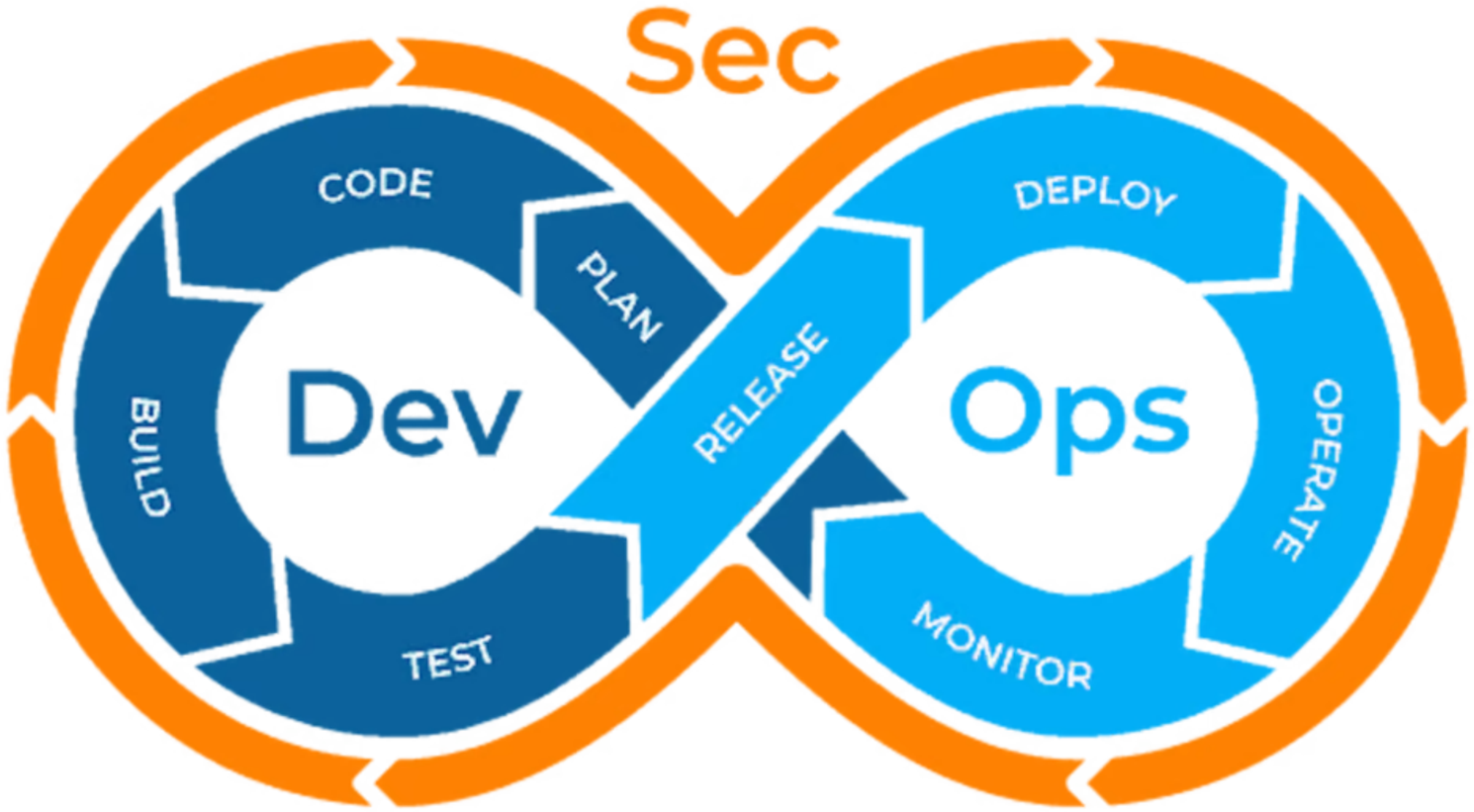
Planned topics for this lesson:

- What is DevSecOps ?
  THE NEED FOR SECURITY IN DEVOPS
- Static code analysis
  STATIC ANALYSIS AND CODE QUALITY CHECKS



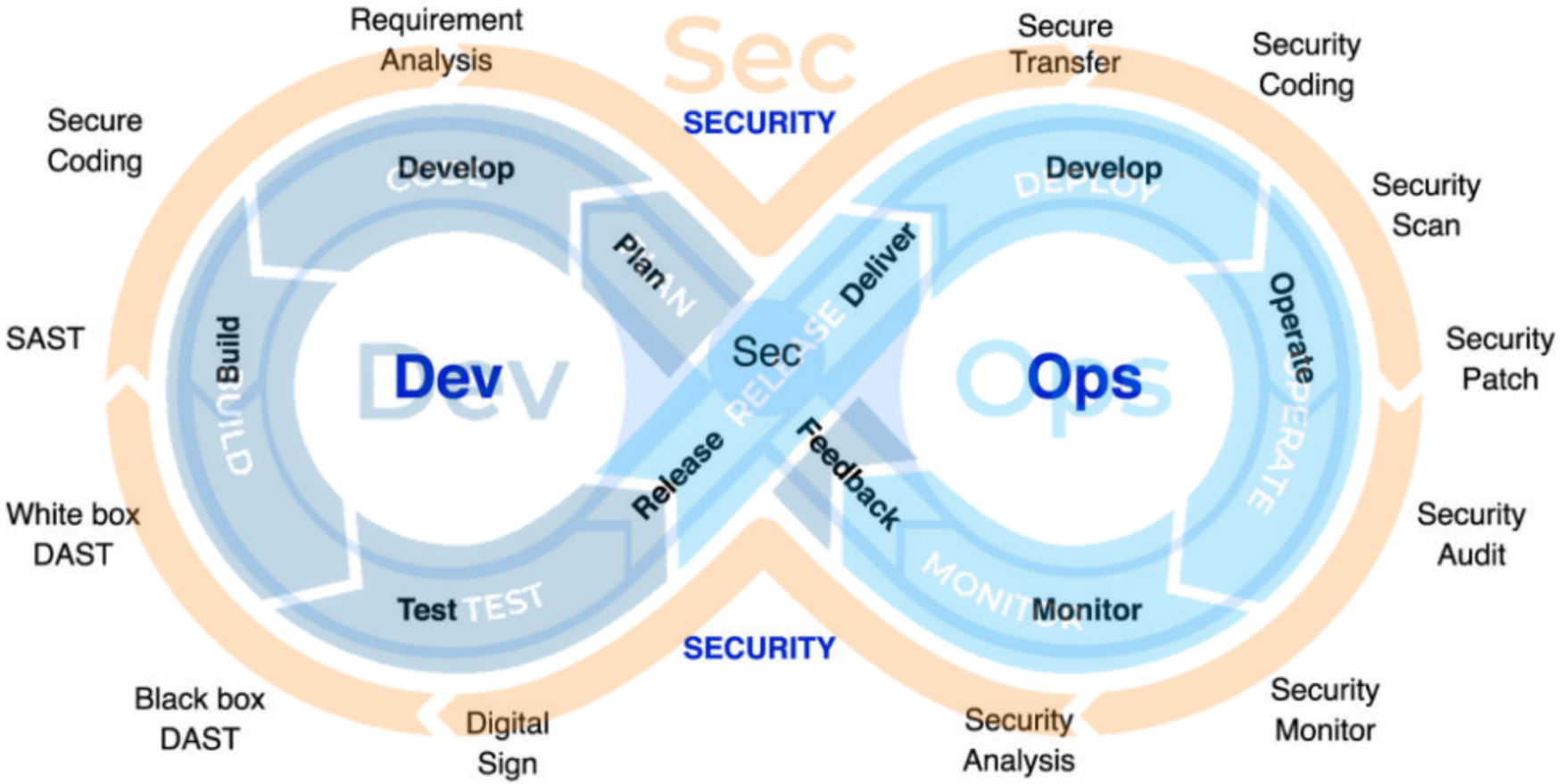
CI/CD Pipeline

Example of a continuous software development system:



CI/CD Pipeline

Example of a continuous software development system:



Security in DevOps

Example of a continuous software development system:

Why Security:

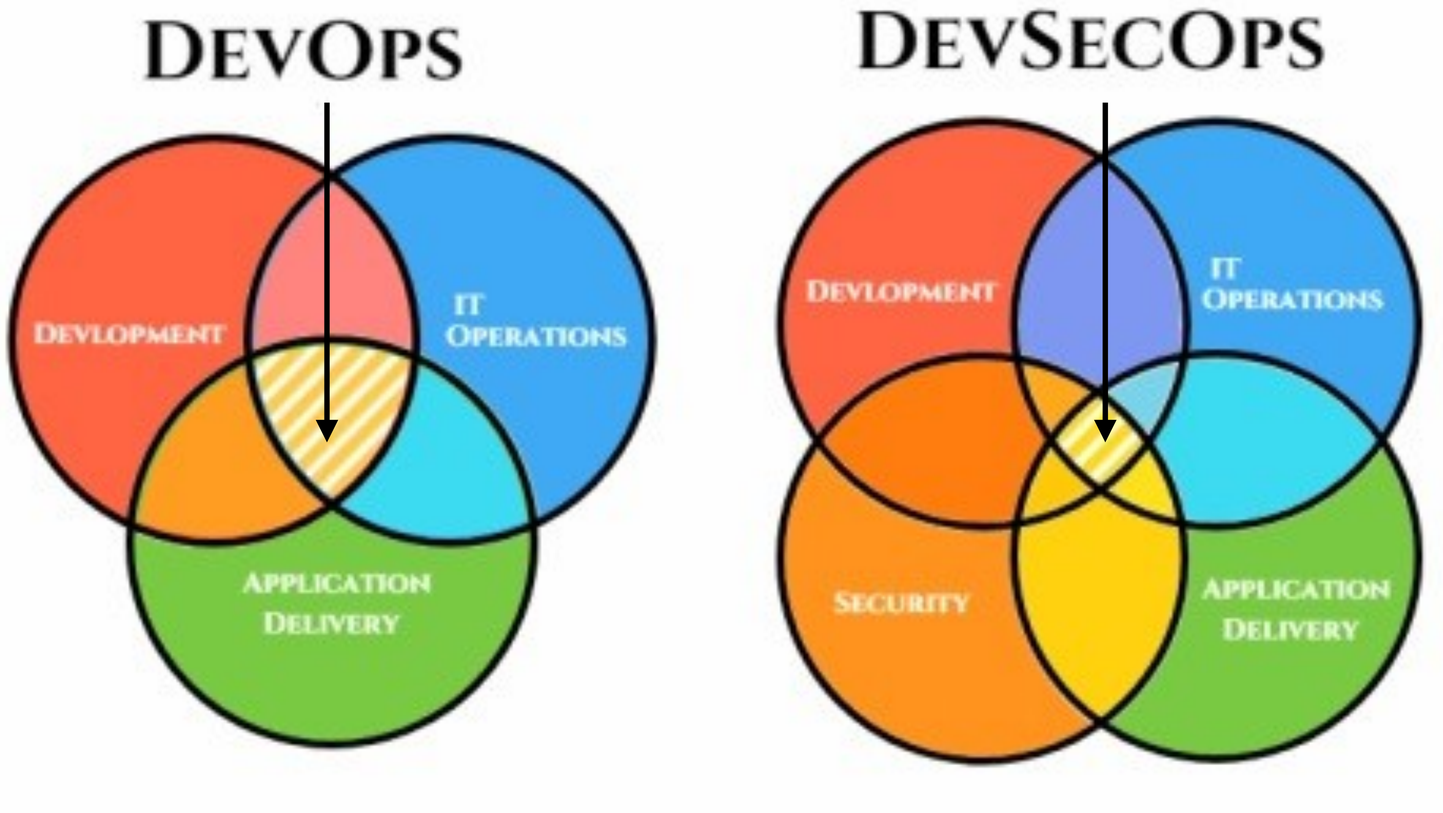
- Traditional development cycles had security at the end, leading to costly vulnerabilities in production.
- Modern applications involve complex microservices, cloud infrastructures, and frequent releases that increase attack surfaces.

Key Risks in Modern Development:

- Faster Development = Higher Risks:
  - Without security baked into the process, vulnerabilities go unnoticed until late stages.
- Complex Architectures:
  - Containerised environments and cloud infrastructure create new attack vectors.
- Increasing Cyberattacks:
  - 2023 saw a rise in supply chain attacks, phishing, and ransomware incidents.

Security in DevOps

Example of a continuous software development system:



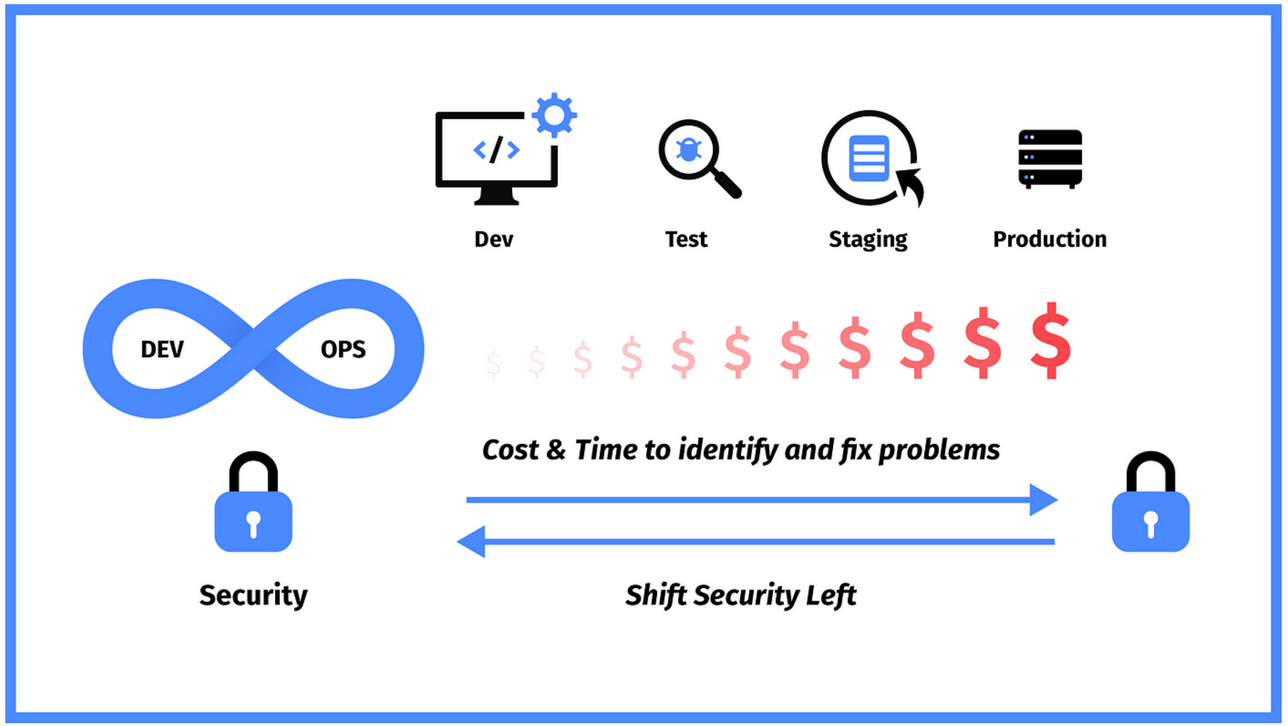
What is DevSecOps?

Shifting Security Left

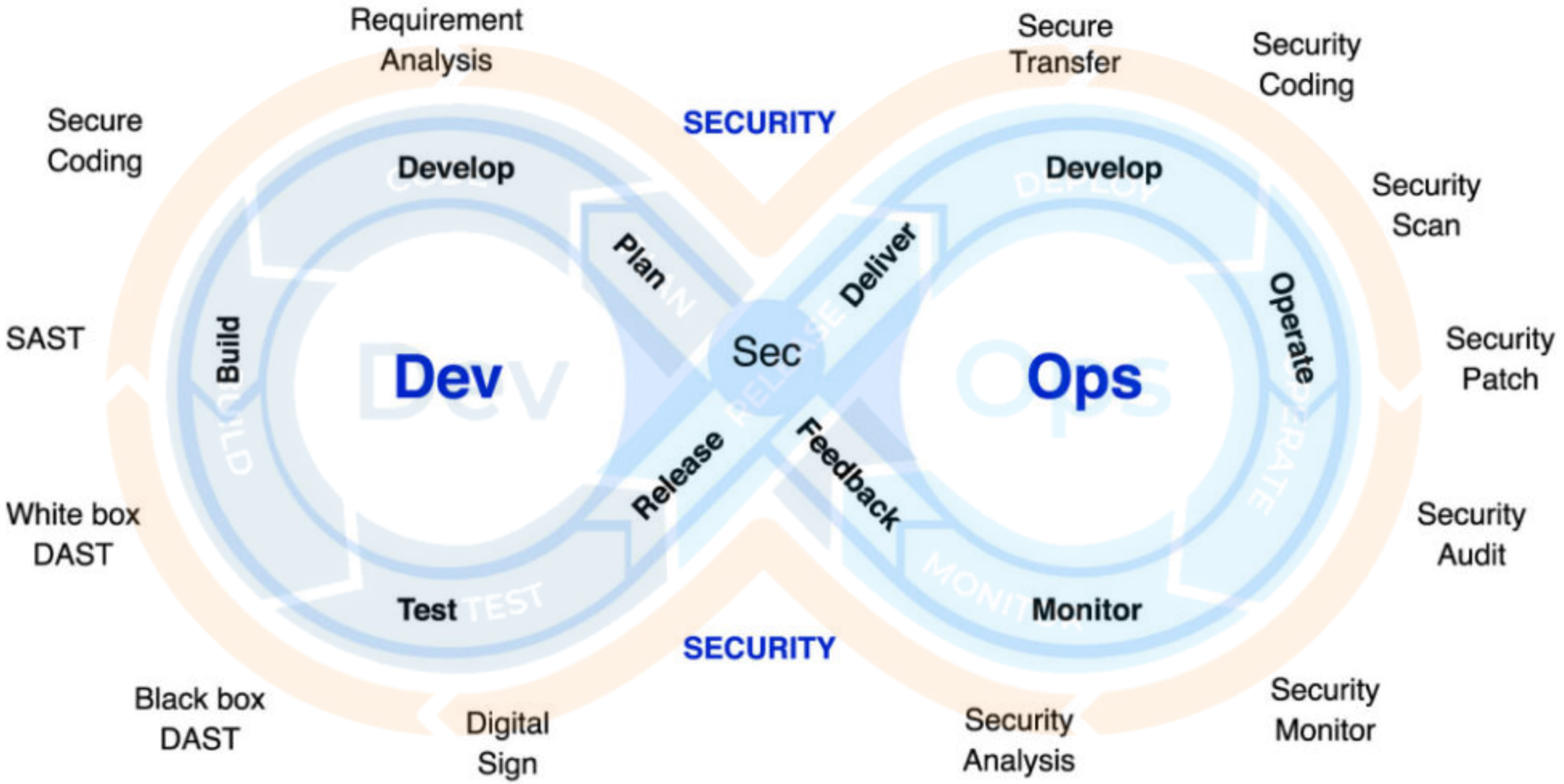
- Integrating security throughout the entire DevOps lifecycle.
- Shift-left security: Moving security practices earlier in the development process to catch vulnerabilities before deployment.

Why Shift Left?:

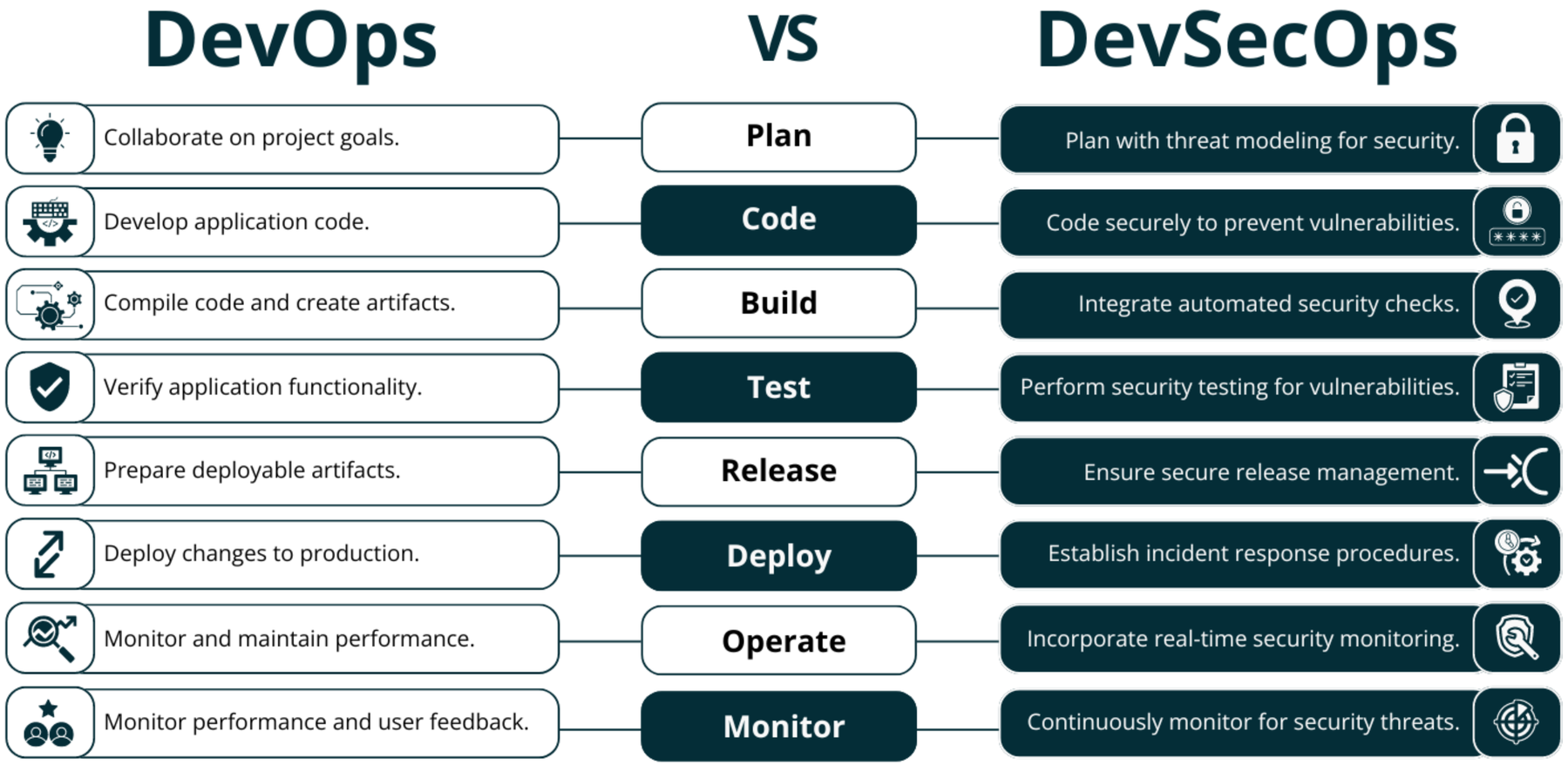
- Detecting vulnerabilities early is cheaper and easier to fix.
- Reduces attack vectors from the start of the development process.
- Real-time visibility into security risks during development, not just post-deployment.



Core Principal of DevSecOps



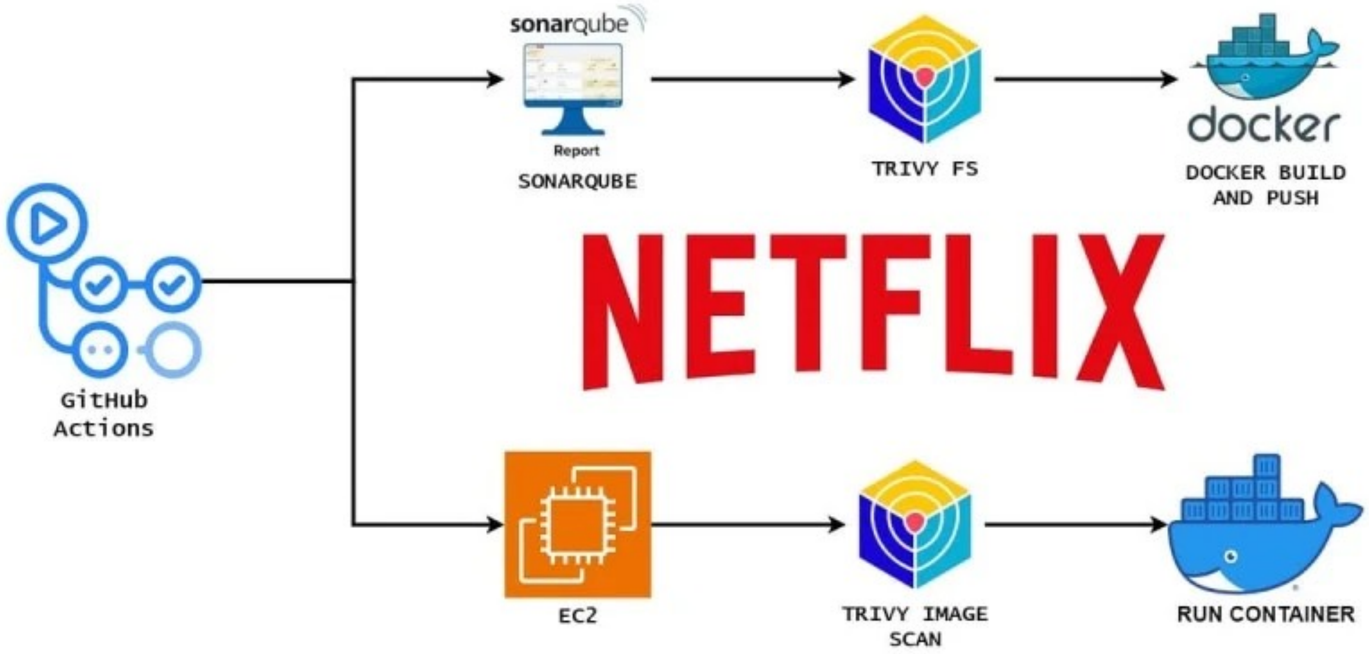
Traditional Security vs DevSecOps



Benefits of DevSecOps

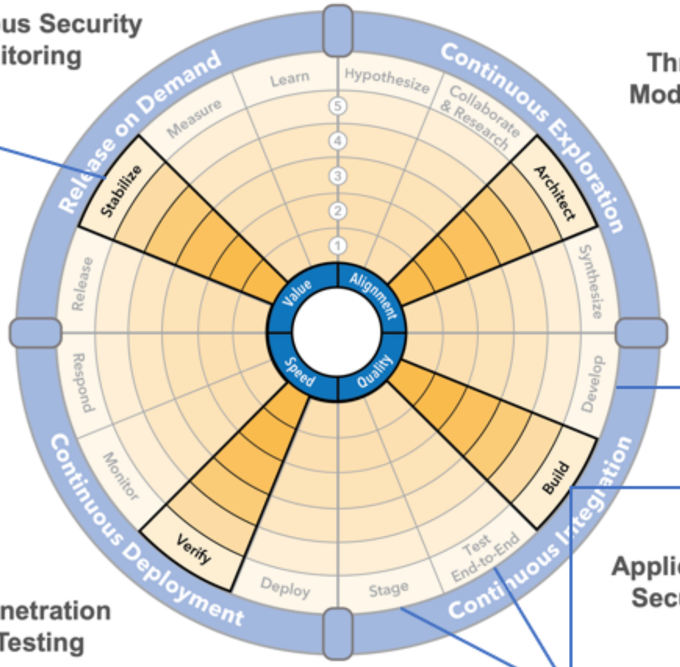
Why Implement DevSecOps?

- Reduced Time to Fix Bugs: Fixing vulnerabilities earlier in development is faster and cheaper.
- Continuous Security: Automated tests and monitoring ensure security across the pipeline.
- Better Compliance: Ensures adherence to industry standards (e.g., GDPR, PCI-DSS) through continuous security checks.
- Improved Collaboration: Security becomes a shared responsibility, promoting teamwork.



Security Challenges

Key Vulnerabilities



Continuous Security Monitoring

Security Response Team
Incident response
Security bulletins
Security mailboxes
White head hacker relations

SECURITY MISCONFIGURATION

Threat Modeling

INSECURE DEPENDENCIES

Security IDE plug-ins
Code reviews/pair work

Static code analysis → INJECTION ATTACKS
Third-party scans
Fuzz testing
Code signing

Application Security

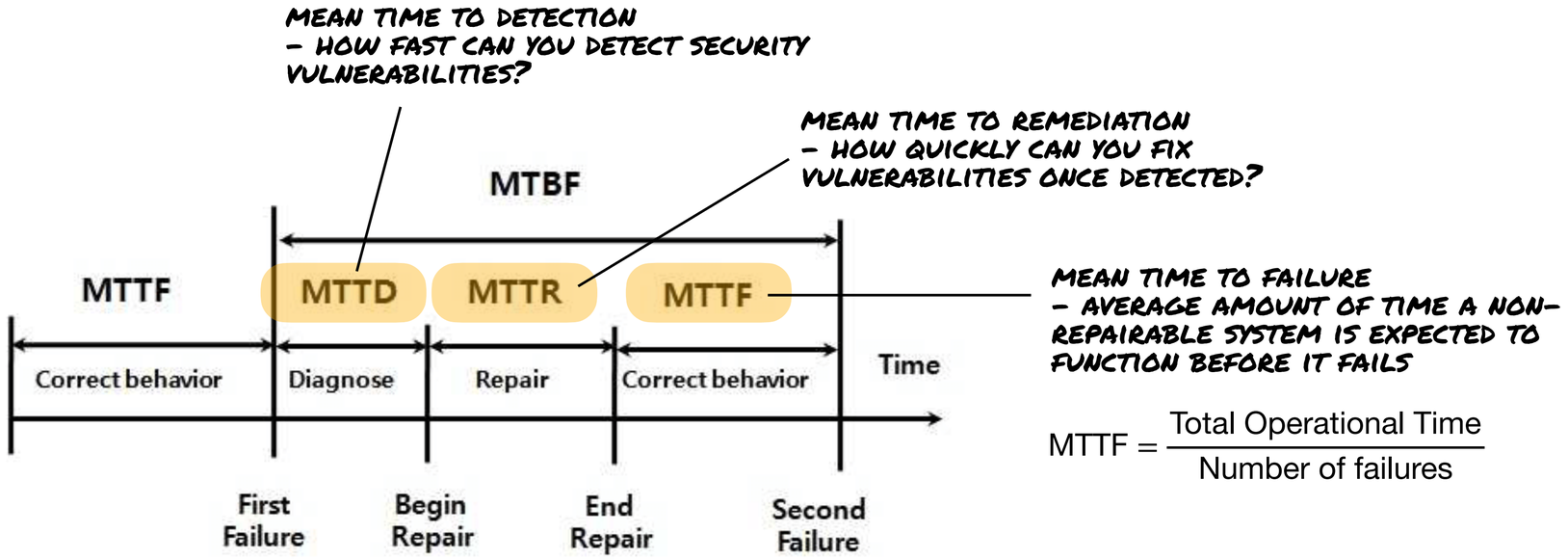
Penetration Testing

Infrastructure scans
Malware scans
Dynamic scans

HUMAN ERROR ?!!

Security Challenges

Key Metrics



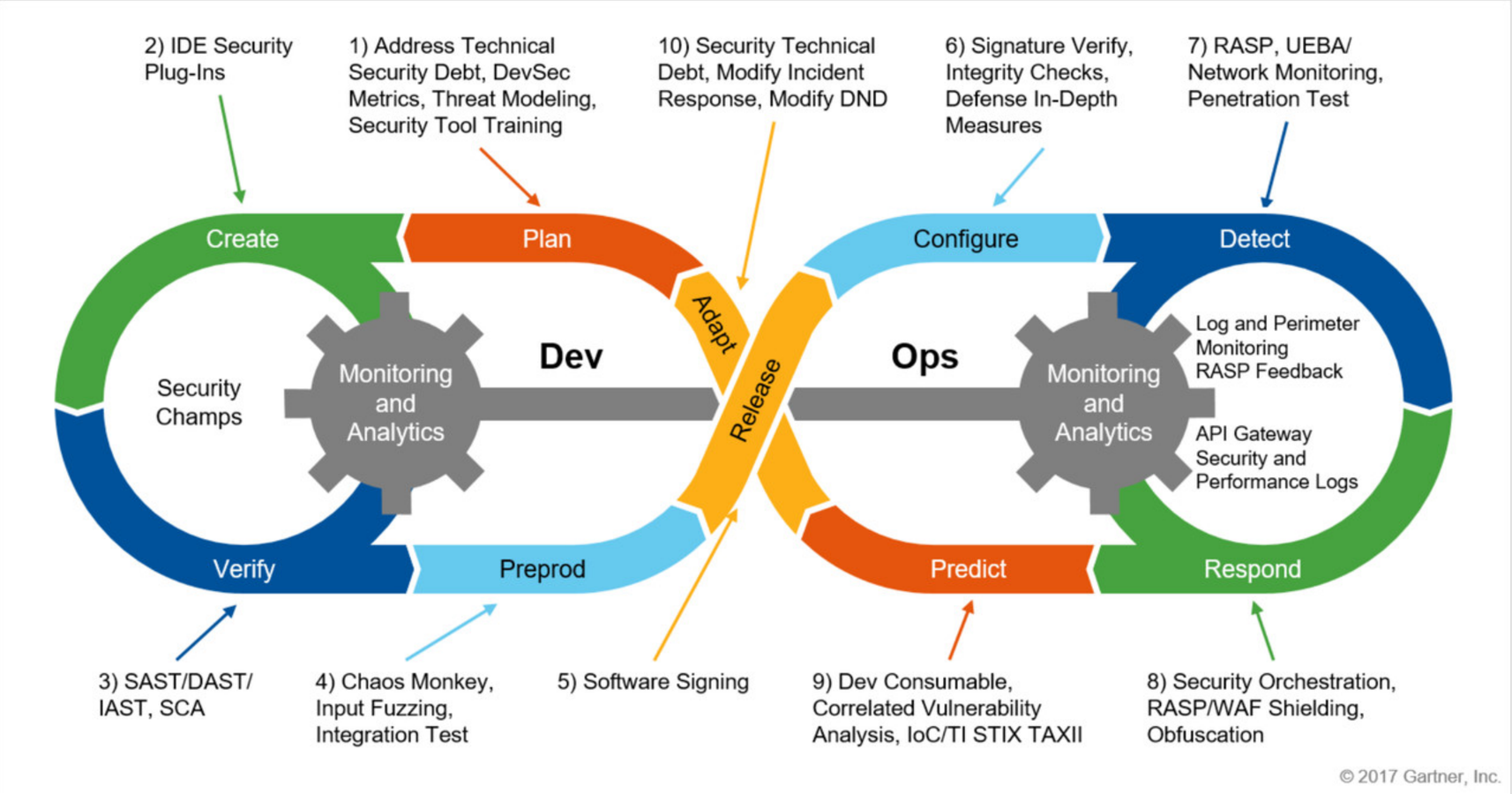
MTBF (Mean Time Between Failures)

- similar metric for repairable systems
- include the time to failure and the time it takes to repair the system

$$MTBF = MTTD + MTTR$$

Security Challenges

Best Practices



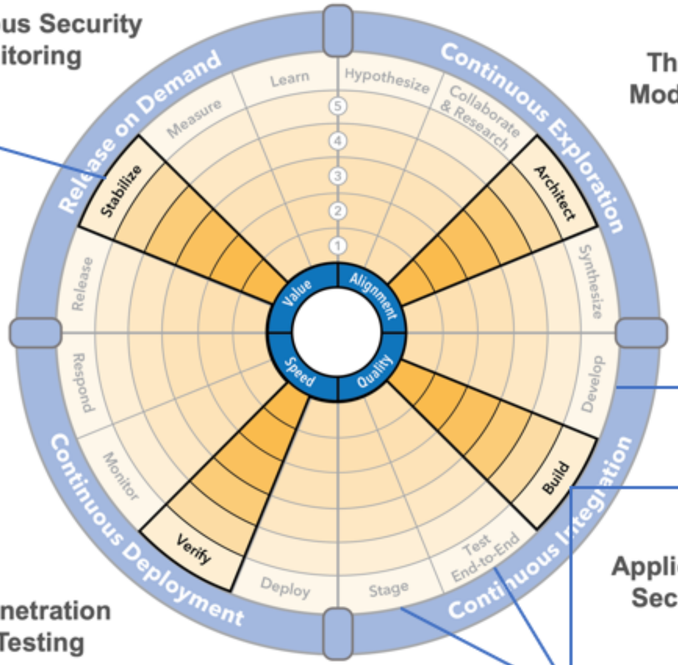
Security Challenges

Best Practices

- **Security as Code**: Treat security policies and tests like code. Use version control, collaboration, and automationReduces attack vectors from the start of the development process.
- **Automated Testing**: Integrate automated security testing into CI/CD pipelines (static, dynamic, and dependency checks).
- **Continuous Monitoring**: Implement tools for real-time monitoring of security events in production.
- **Infrastructure as Code (IaC)**: Automate secure configurations of infrastructure to avoid security misconfigurations.
- **Training and Awareness**: Regularly train teams on the latest security practices and vulnerabilities.

Security Challenges

Key Vulnerabilities



Continuous Security Monitoring

Security Response Team
Incident response
Security bulletins
Security mailboxes
White head hacker relations

SECURITY MISCONFIGURATION

HUMAN ERROR ?!!

Threat Modeling

INSECURE DEPENDENCIES

Security IDE plug-ins
Code reviews/pair work

Static code analysis → INJECTION ATTACKS
Third-party scans
Fuzz testing
Code signing

Application Security

Penetration Testing

Infrastructure scans
Malware scans
Dynamic scans