Sisteme distribuite

Mihai Zaharia

Cursul 13

Reziliența unei aplicații distribuite

- Aceasta descrie măsura în care un sistem/infrastructură digitală sau arhitectură a unei aplicații este capabilă să își mențină o corectă funcționare a serviciilor oferite
- refacerea automată
- Percepție greșită

Reziliența încotro?

- Nivel de infrastructura -> la nivelul aplicațiilor IT
- Momentan cloud = acceptabil
- Eficiența în cyberspace
- Agilitate aplicației ?

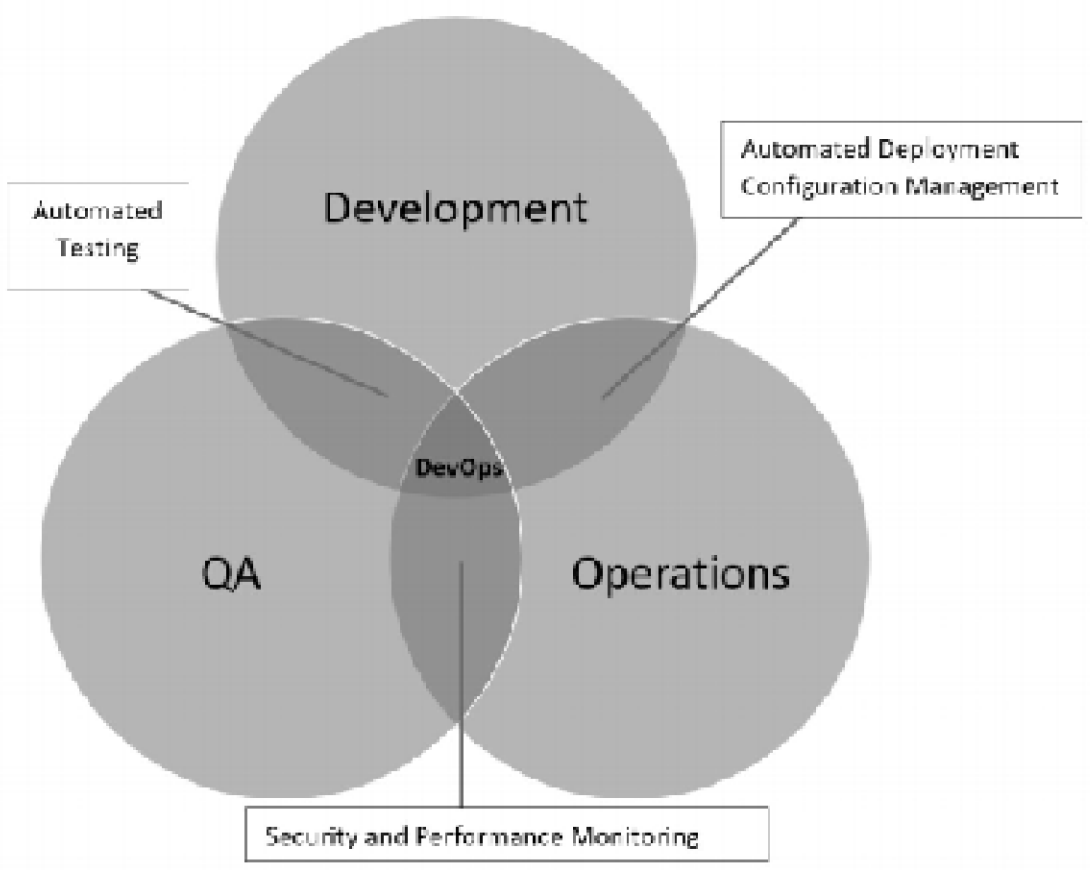
Reziliența încotro?

- Reducerea costurilor de restaurare automată
- Testarea?
- Reziliența distribuită?
- Clasic -> distribuit = problemă

Rezilienţa observaţii

- Abordările clasice - nu merg
- Chief Information Officer trebuie implicat
- Chiar toate cursurile clasice nu mai trebuie???

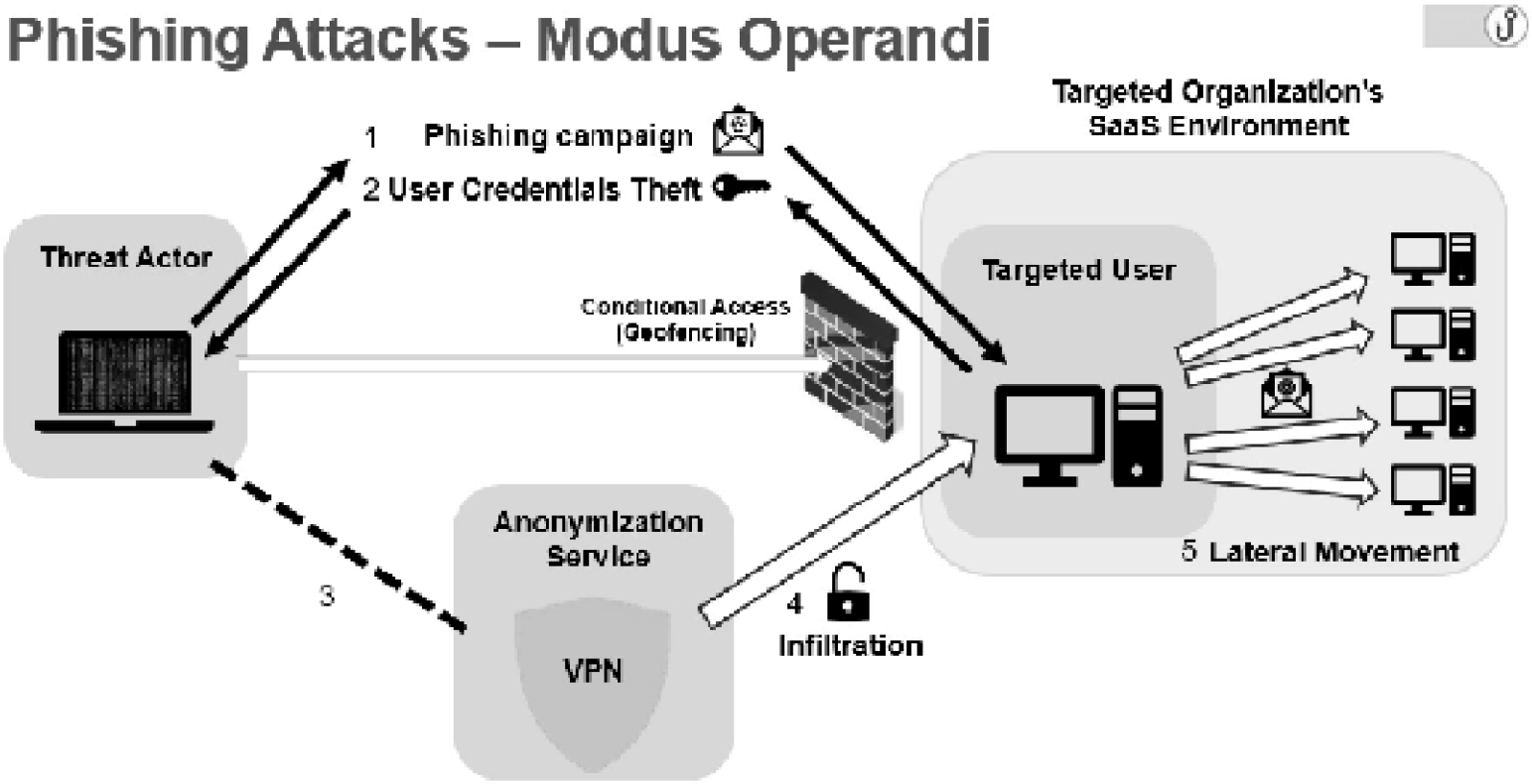
DevOps


Development
Automated Deployment
Configuration Management
Automated
Testing
DevOps
QA
Operations
Security and Performance Monitoring

SecOps

Răspândirea necontrolată a mașinilor virtuale

Pescuitul în nori



Ce este un siloz de date?

- Cauze
  - culturale
  - structurale
  - tehnologice

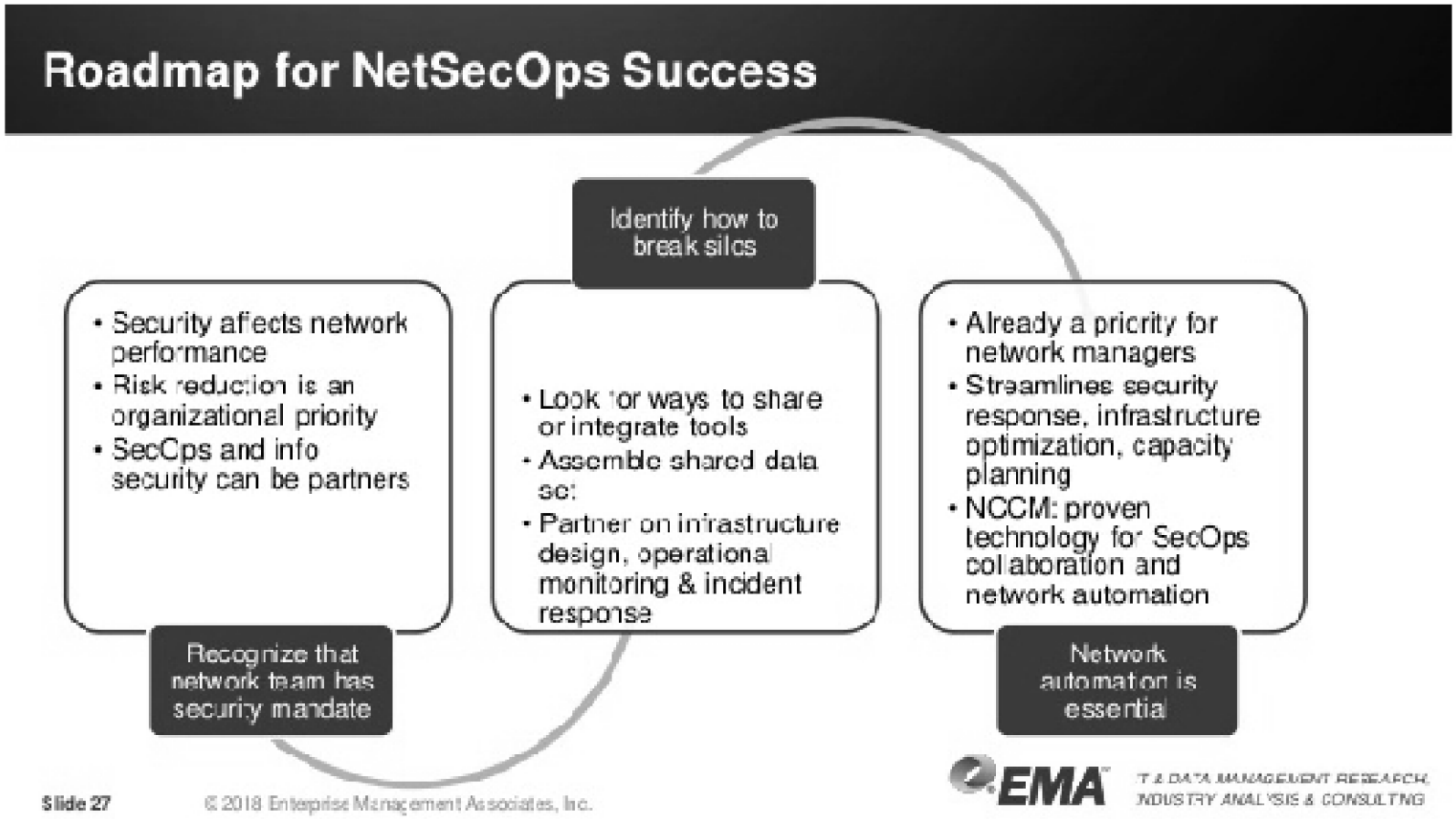
Sunt bune silozurile de date?

- x

Proști daaaa mulți ....

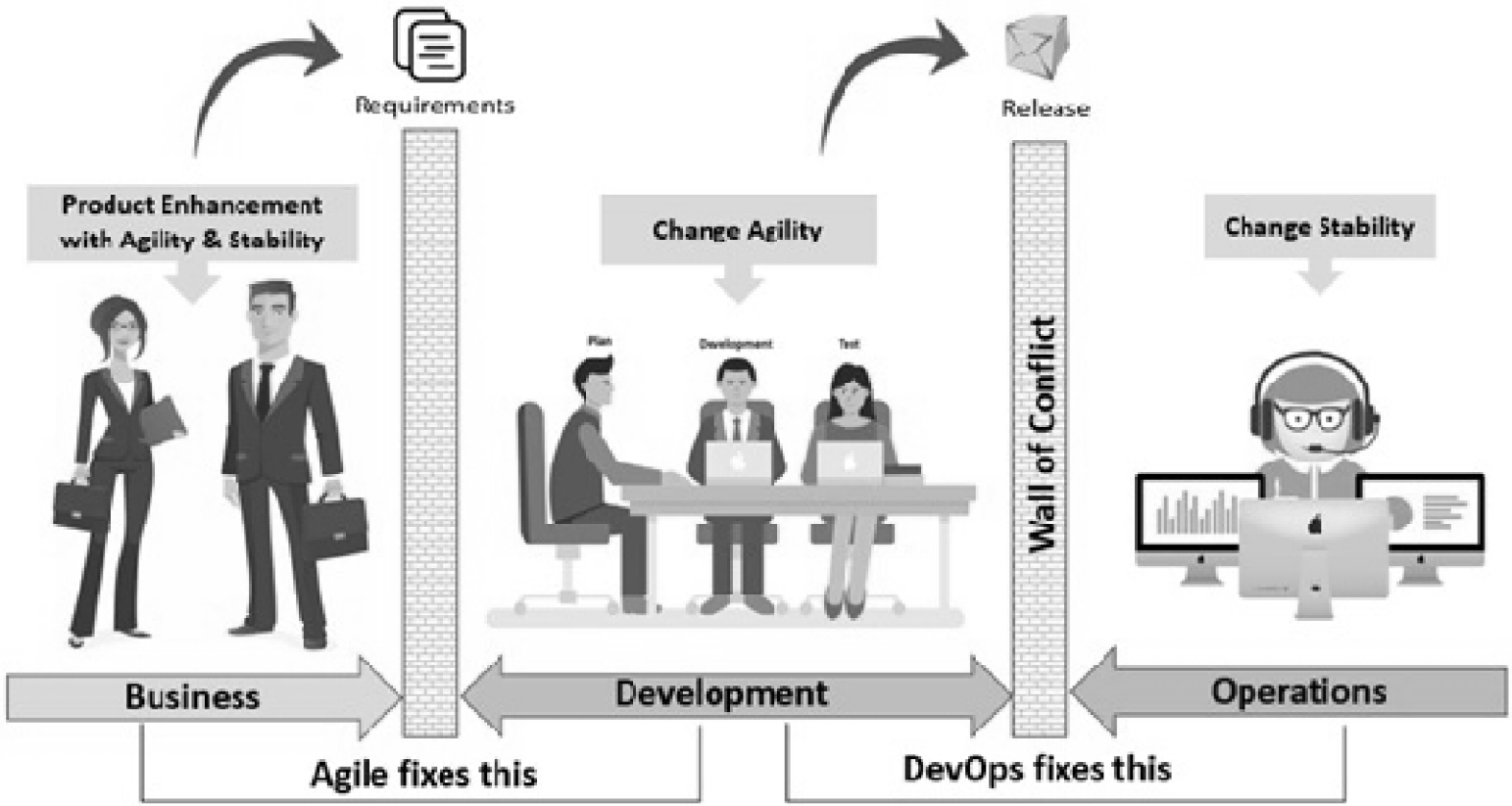
- x

Silozurile și NetOps, SecOps



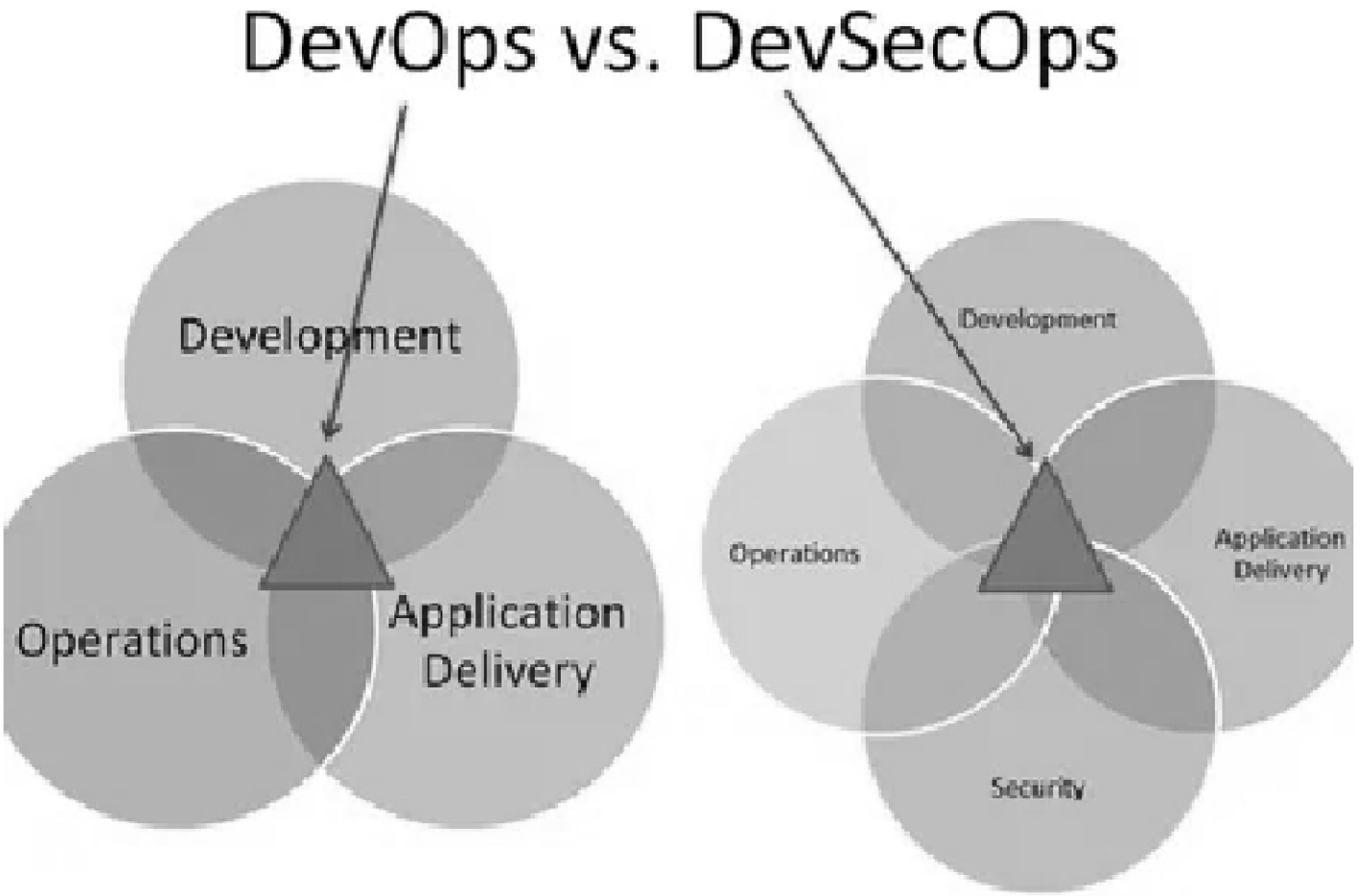
- X

Daca se combina Agile cu DevOps atunci...



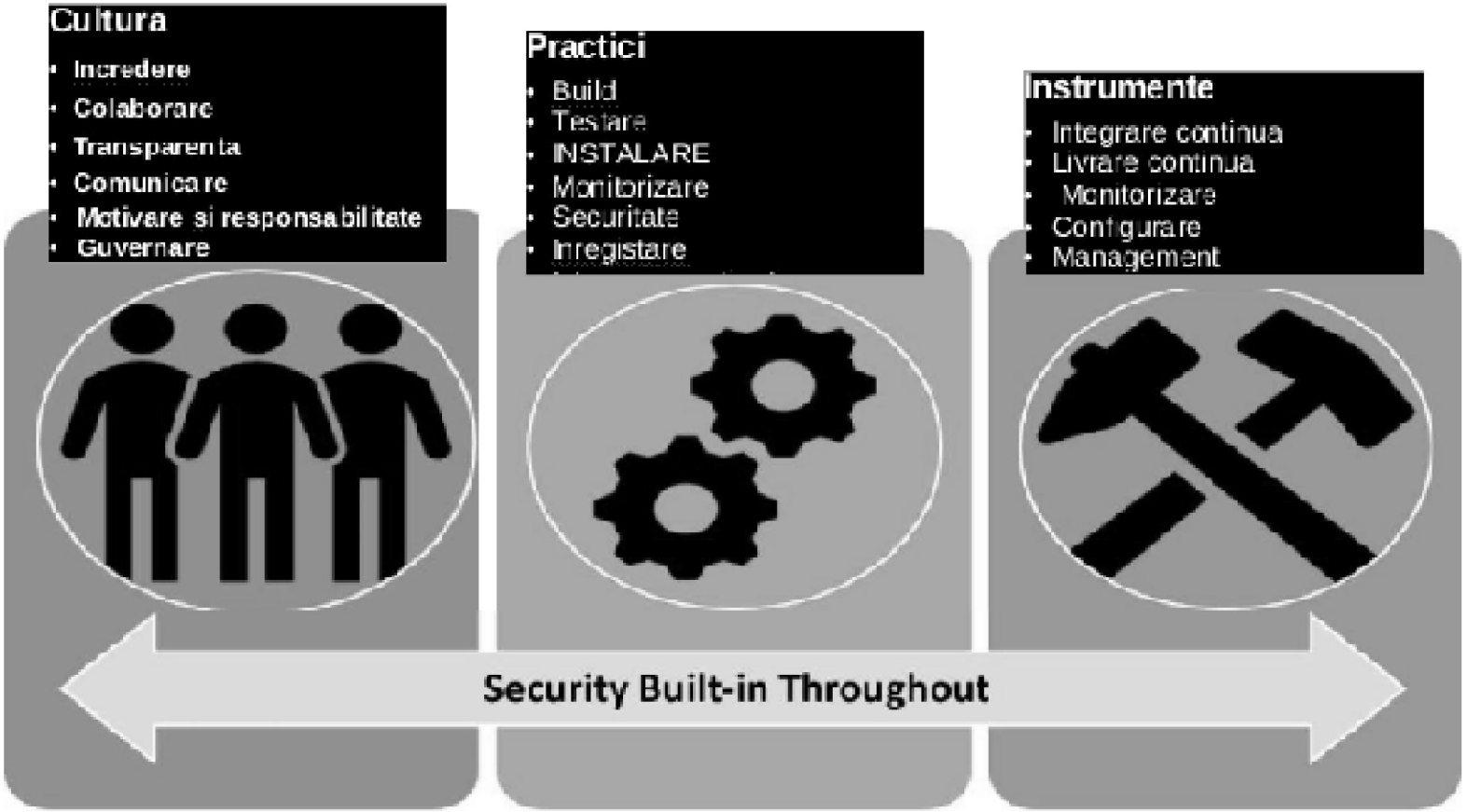
- X

DevSecOps?


DevOps vs. DevSecOps
Development
Operations
Application
Delivery
Development
Operations
Application
Delivery
Security

- X

Dimensiunile DevSecOps

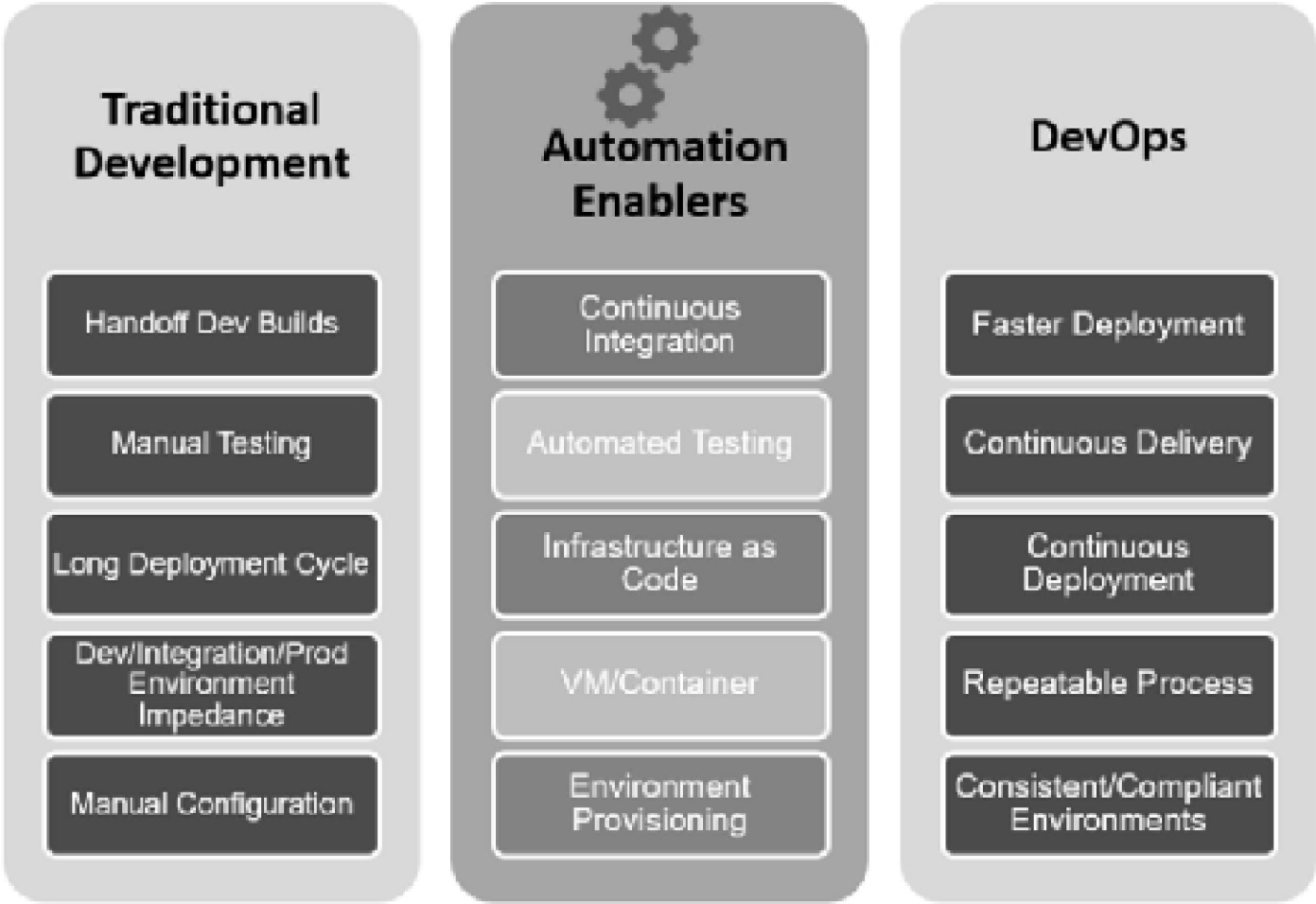


- X

De ce ar fi mai bun DevSecOps?

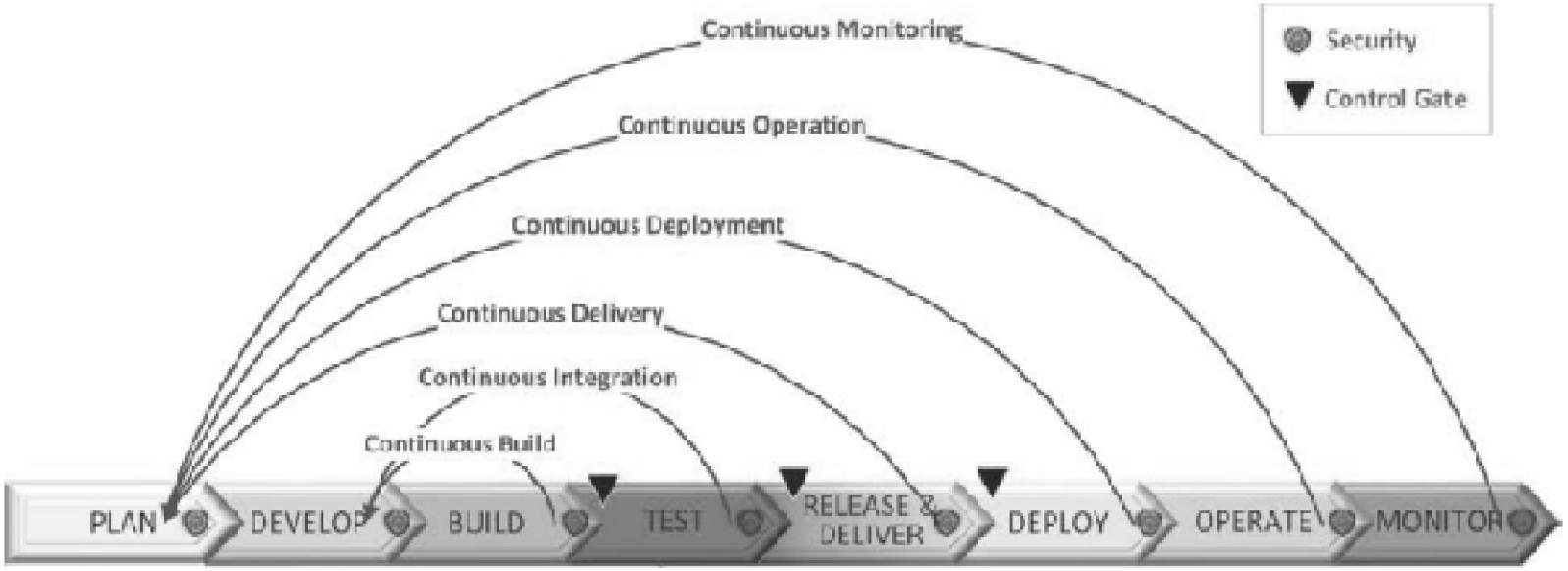
Problemele specifice ciclului clasic de dezvoltare	Beneficiile oferite de catre DevSecOps
un proces manual repetitiv	configurarea și instalarea automată a aplicației
dezvoltarea durează zile sau săptămâni	instalarea durează câteva minute
nu este repetabil și este supus erorilor	este un proces continuu și repetabil
intervenția umană conduce la inconsistențe	rezultatul este consistent
timpi de cădere/pierduți frecvenți	timpi de cădere minimi
este mai ușoară - trebuie oameni mai puțin antrenați	mai complicat trebuie experți
echipele lucrează în silozuri	colaborare continuă între echipe/dept etc
testarea incipientă/primară de securitate nu este efectuată asupra codului	testare de securitate automată încă din faza de scriere a codului

De ce ar fi mai bun DevSecOps?



- X

Cum se aplică ?



DevSecOps SWOT - Pro

- •Viteza si reproductibilitatea rezultatelor
  - Automatizare
  - Imbunatatire continua
- Agilitate
  - Agile
  - Elimina din problemele Agile

DevSecOps SWOT - Contra

- Securitate și testare
- Controlul și proprietatea la nivel fizic
- Responsabilități comune
- Arhitectura sistemului

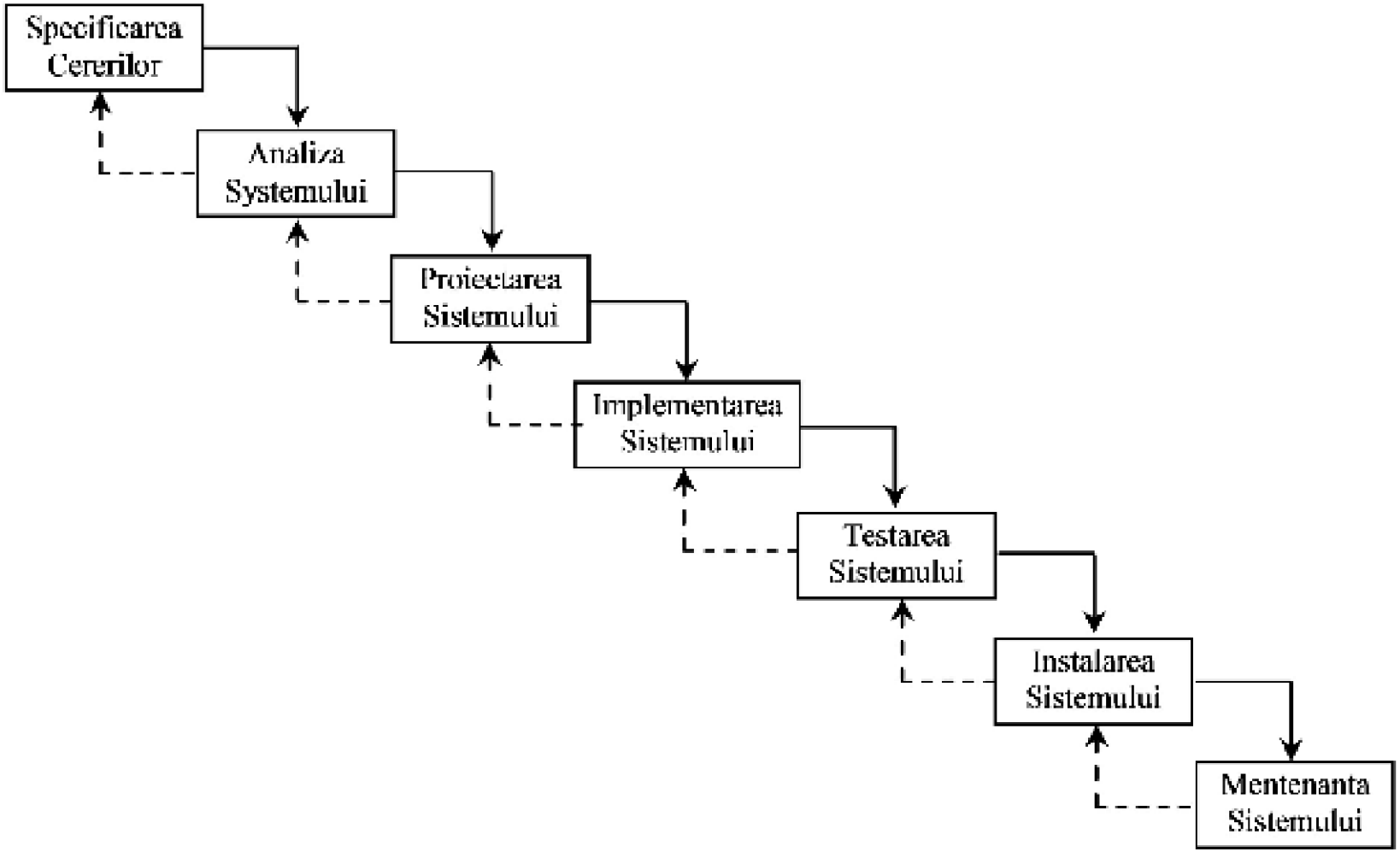
Observatii privind securitatea

- În fiecare etapă din procesele devSecOps
- Din proiectarea aplicației
- Monitorizare continuă

Securitate vs. Testare

Securitate	**Testare**
•procesele de securitate	• evenimentele din teste • mediile de testare
•instrumentele de securitate	• instrumentele de testare
• securitatea accesului (de ex, la mediile DevSecOps)	• testarea accesului (de ex la mediul DevSecOps)
•Vizibilitatea instrumentelor de/pentru securitate (de ex, de-a lungul lantului de dezvoltare (pipeline)	• datele de test
• rapoartele de securitate	• raportarea testarii

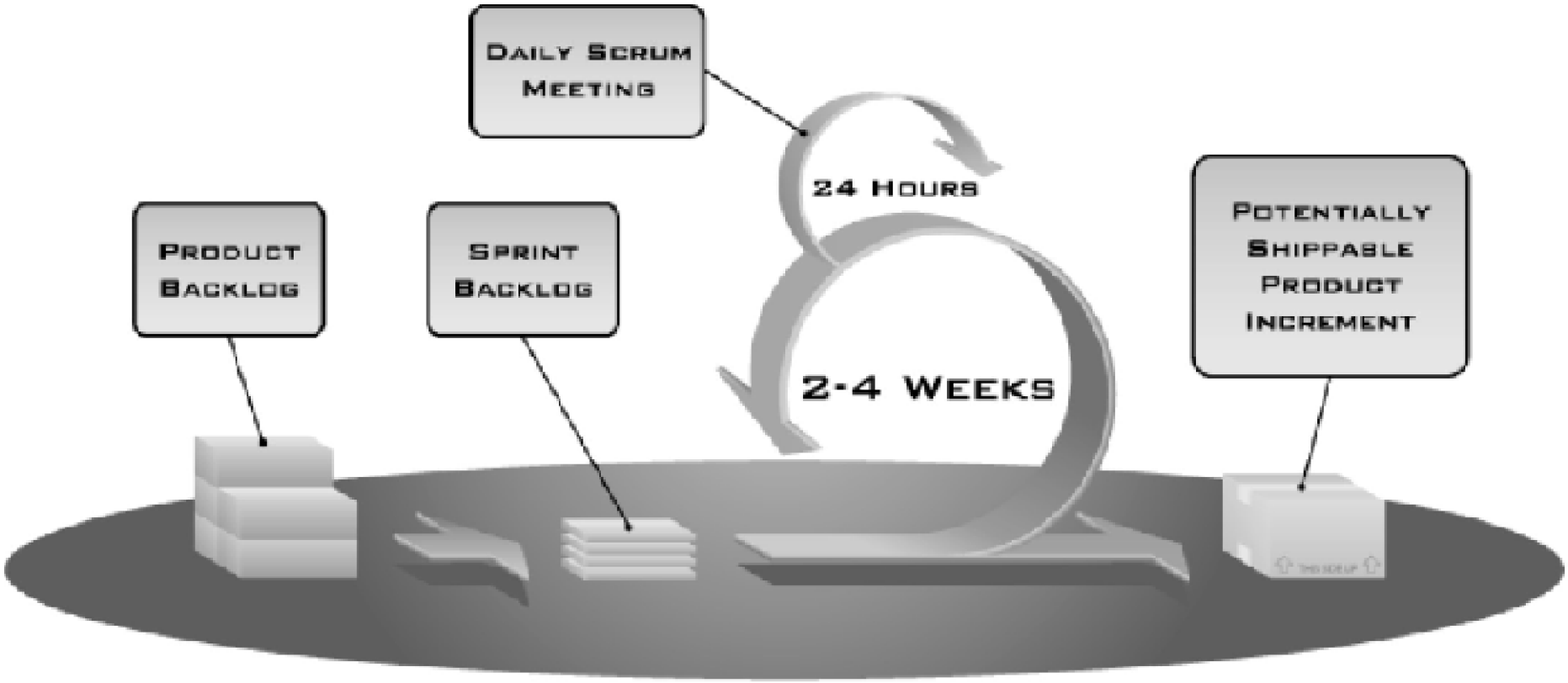
Modelul Cascada - Waterfall



Agile - Beneficiarul

Agile - Echipa

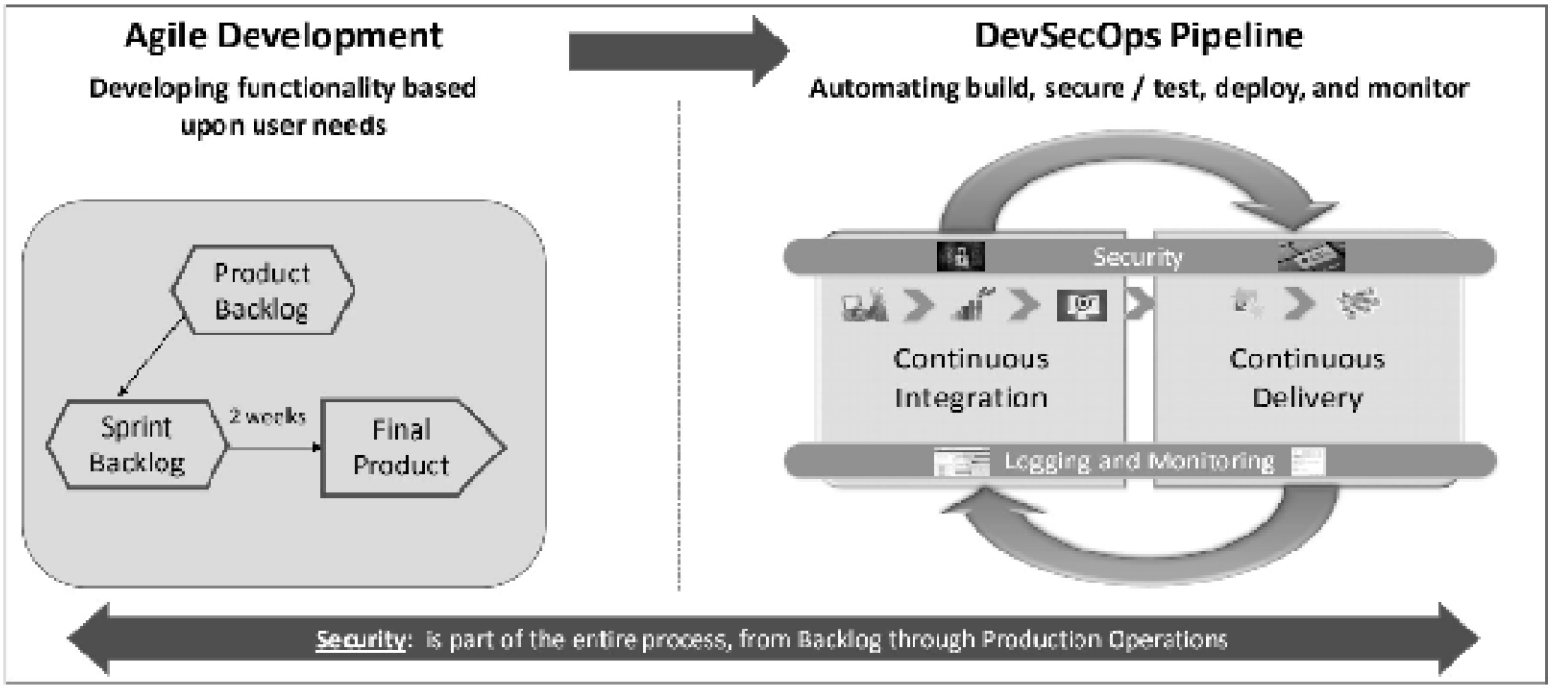
Cum este să fii AGItat(L)



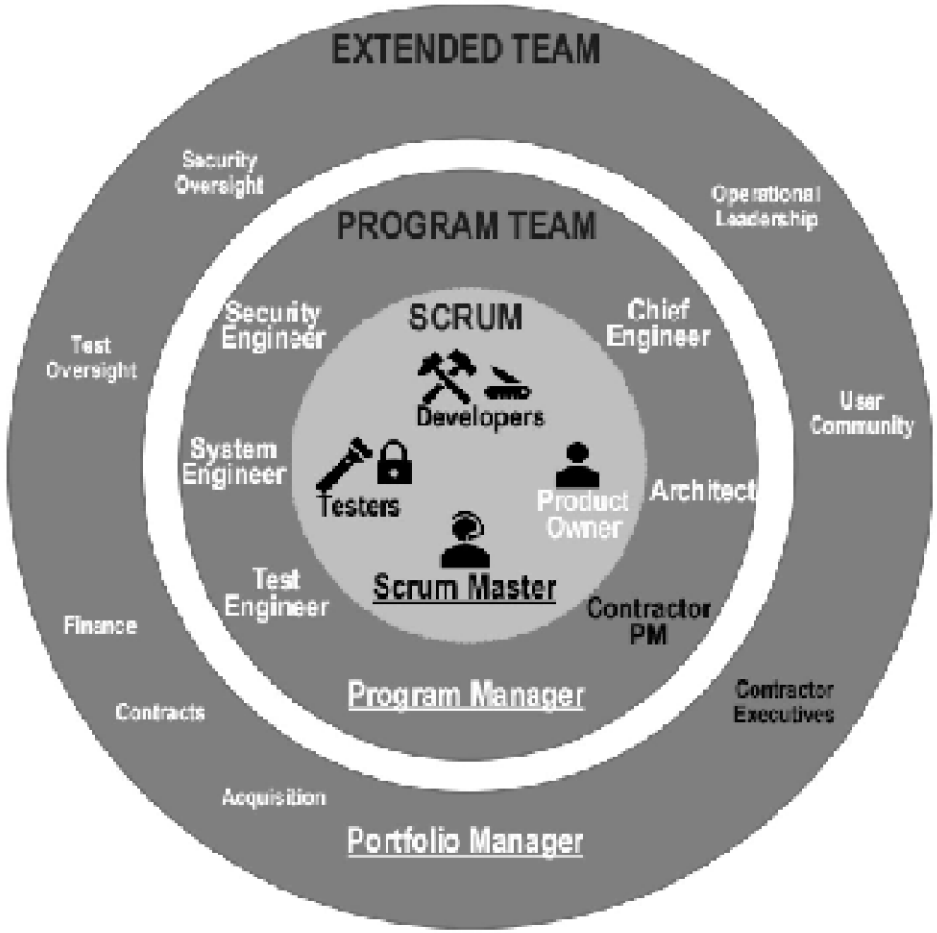
Agile - Sprintul!

Stăpânul SCRUM-ului

Agile + DevSecOps Pipeline



DevSecOps & Agile Scrum



Others Stakeholders:

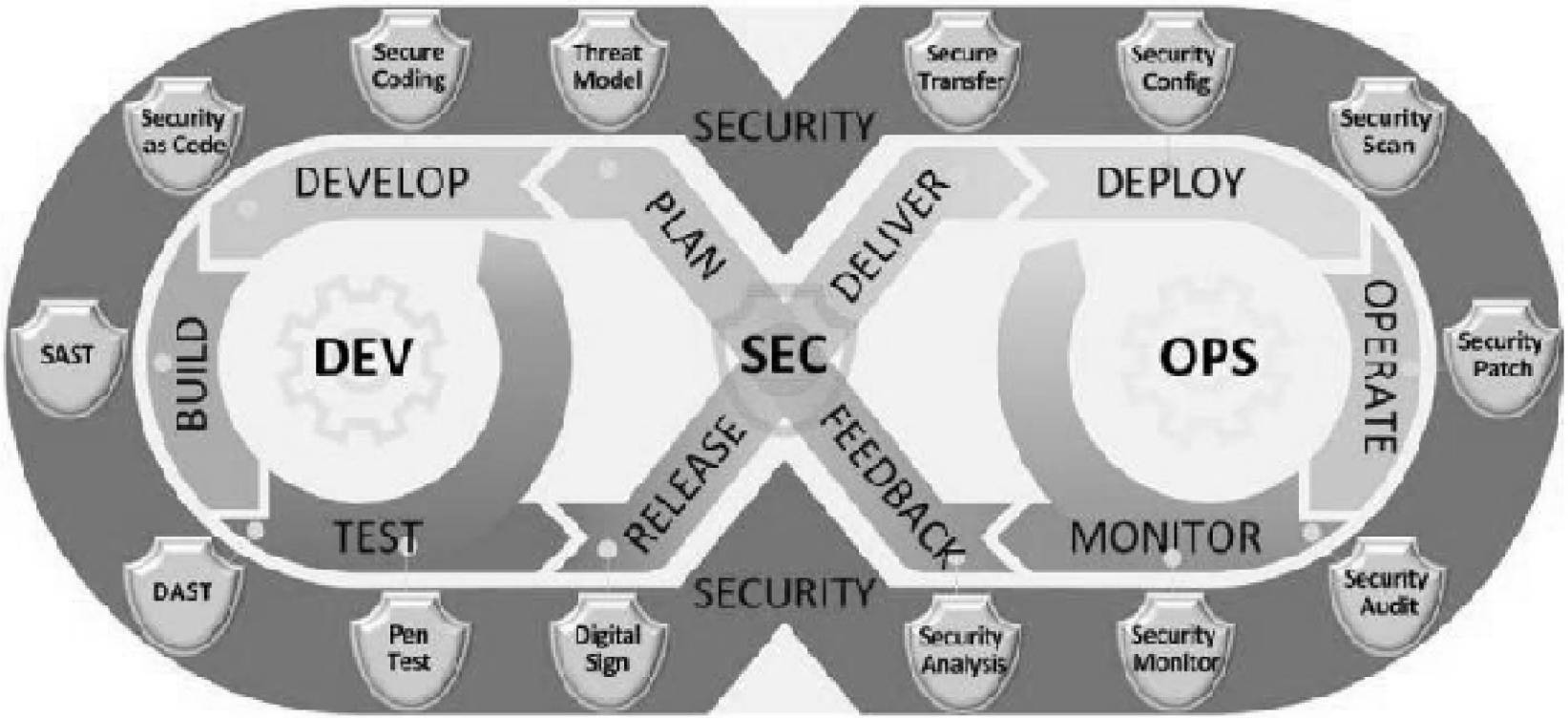
- Operators
- External System Owners and Operators
- External Government or Commercial Data Providers / Consumers
- IT Administrators
- Cloud Service Providers
- Security: CASBs, NSSPs
- SETA Contractors
- FFRDCs

Legend:
Team Name / Level – Blue Font
Government – White Font
Contractor – Black Font
Leader – Underline Font

DevSecOps & Agile Scrum

- Testarea contractorului
- Testare guvernamentală.
- Medii de testare.
- Clarificarea rolurilor

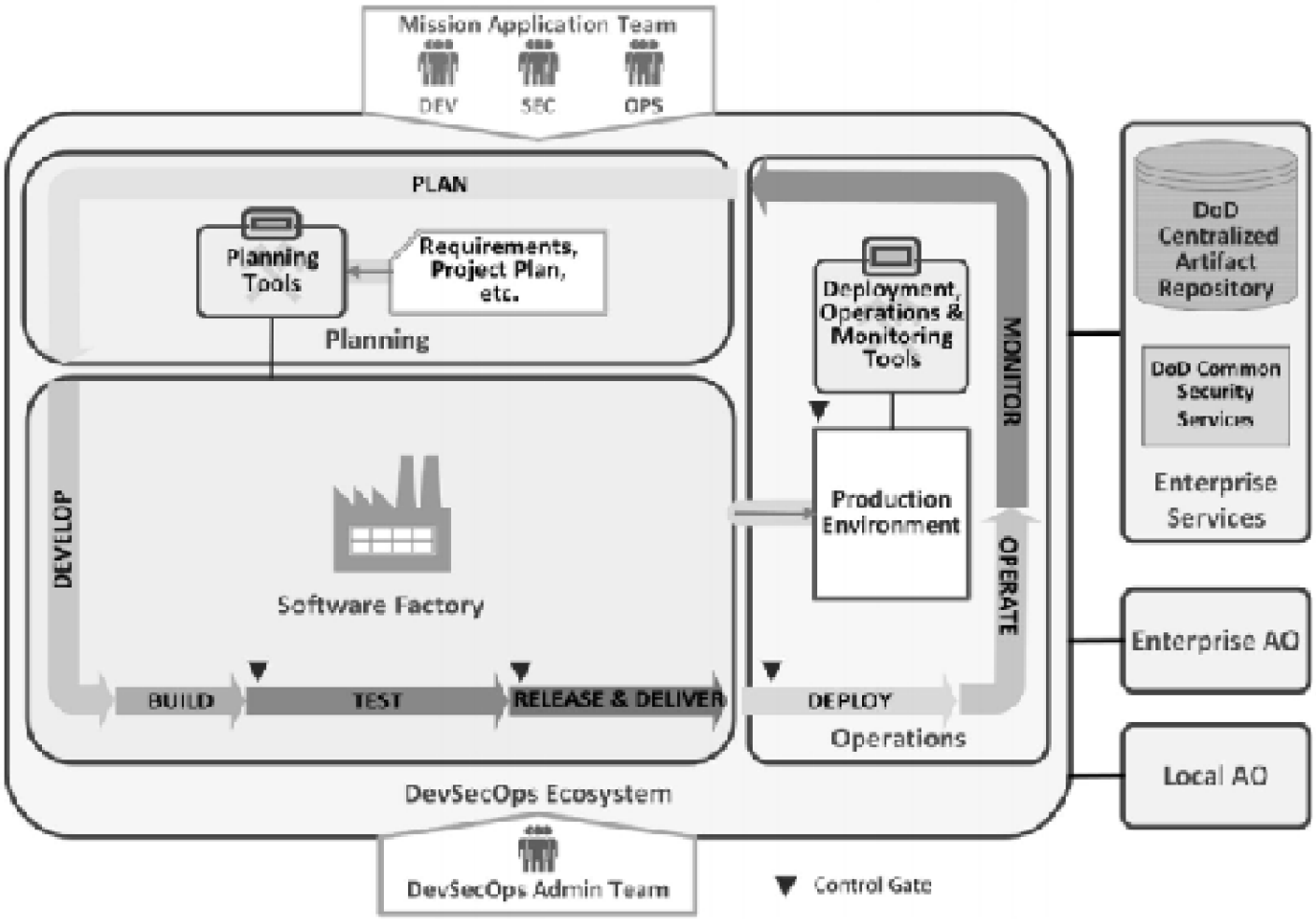
Ciclul de viață al DevSecOps



Pilonii DevSecOps

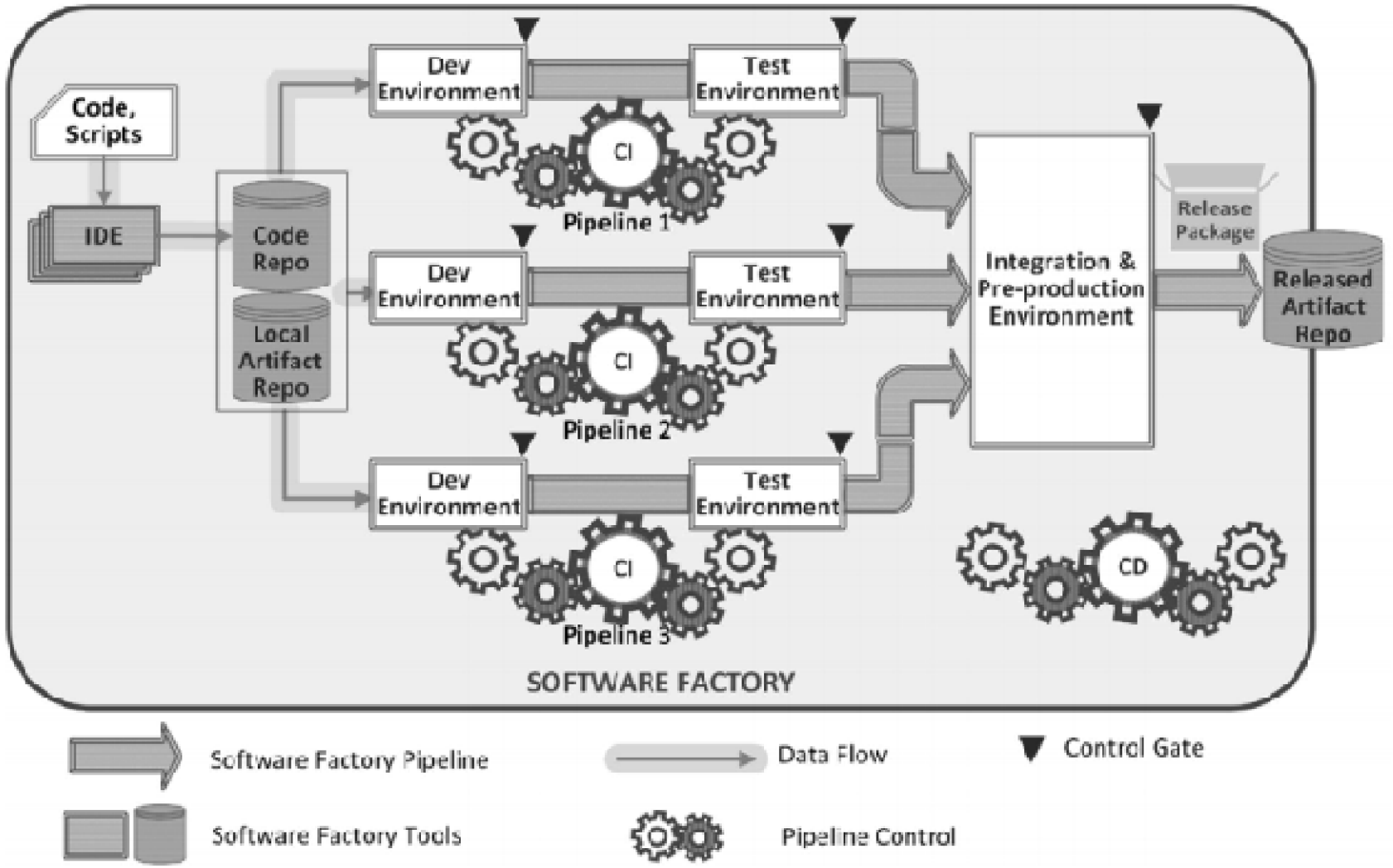
ORGANIZATION	PROCESS	TECHNOLOGY	GOVERNANCE
❖ Culture shift & buy-in ❖ Communication & collaboration ❖ Security/QA throughout ❖ Learn from success/ failure ❖ Feedback and user-driven change	❖ Collaborative design ❖ Test-driven development ❖ Common and automatable tasks ❖ Continuous adaptation and improvement ❖ Continuous ATO	❖ Tool adoption ❖ Automation and orchestration ❖ Cloud and containerization ❖ Infrastructure as Code ❖ Security as Code	❖ Built-in governance control ❖ Uniform policy enforcement ❖ Data-driven validation ❖ Enhanced visibility ❖ Inherited certifications and authorizations

Ecosistemul DevSecOps

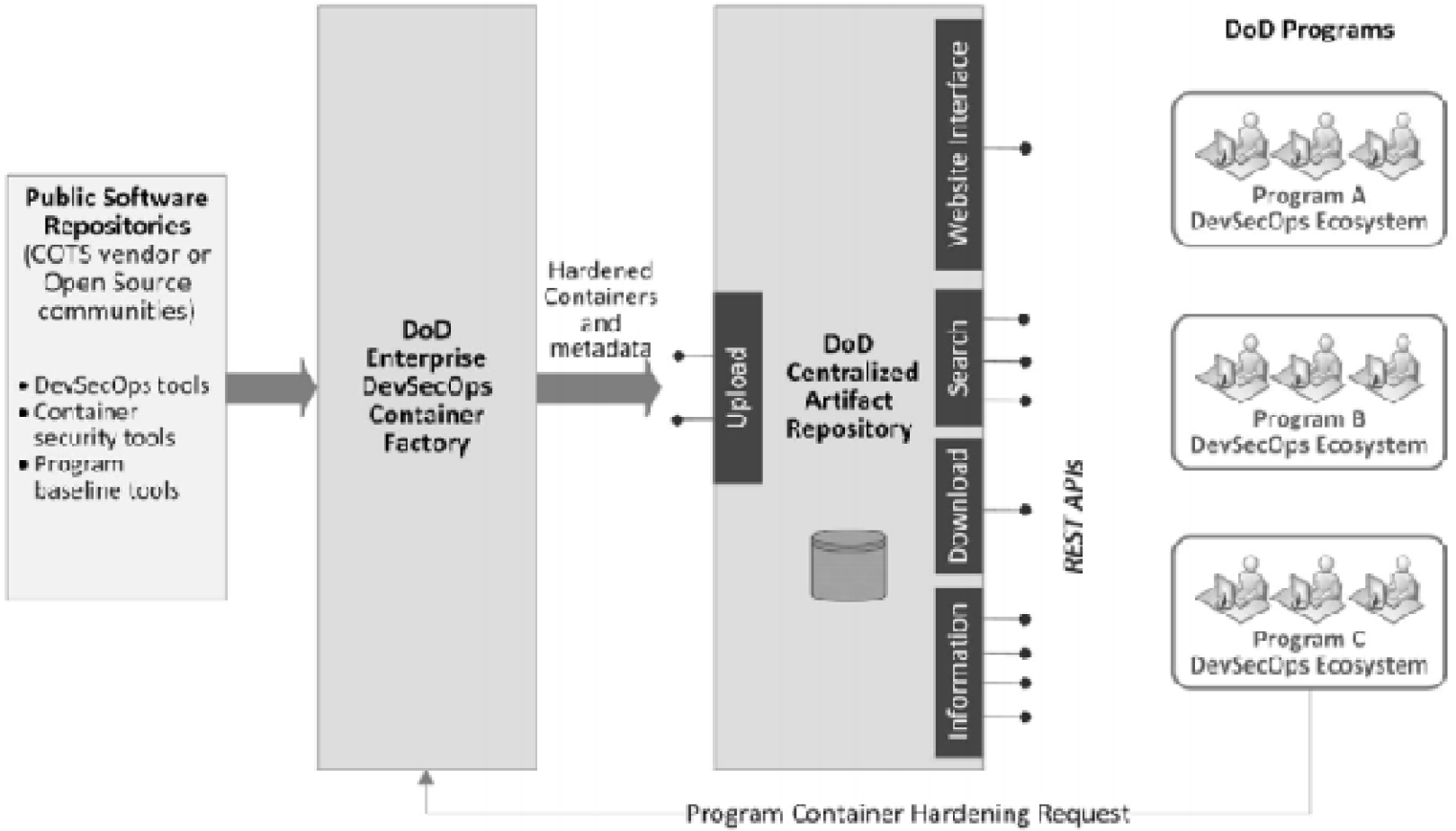


- X

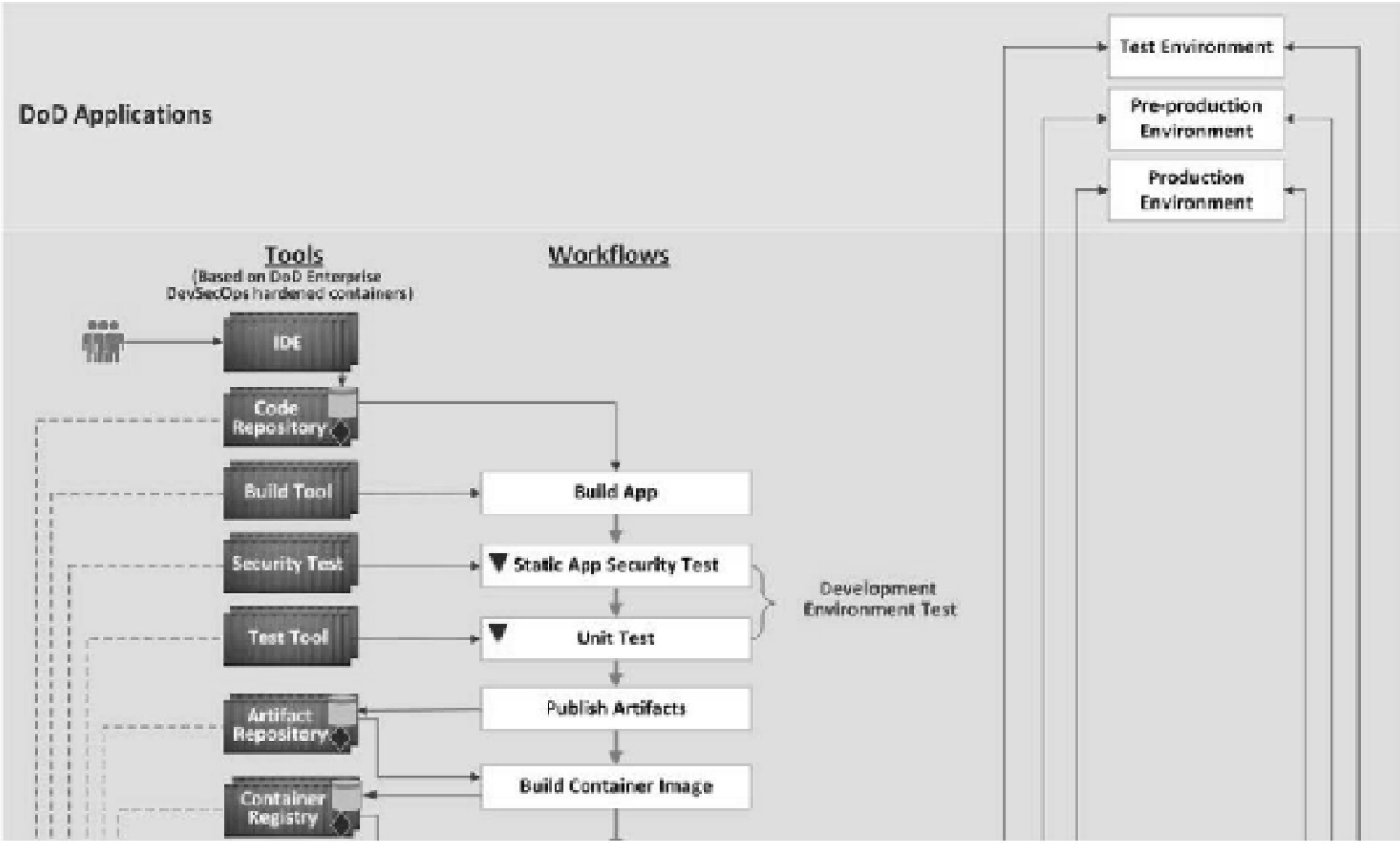
Fabrica software



Serviciul de containere

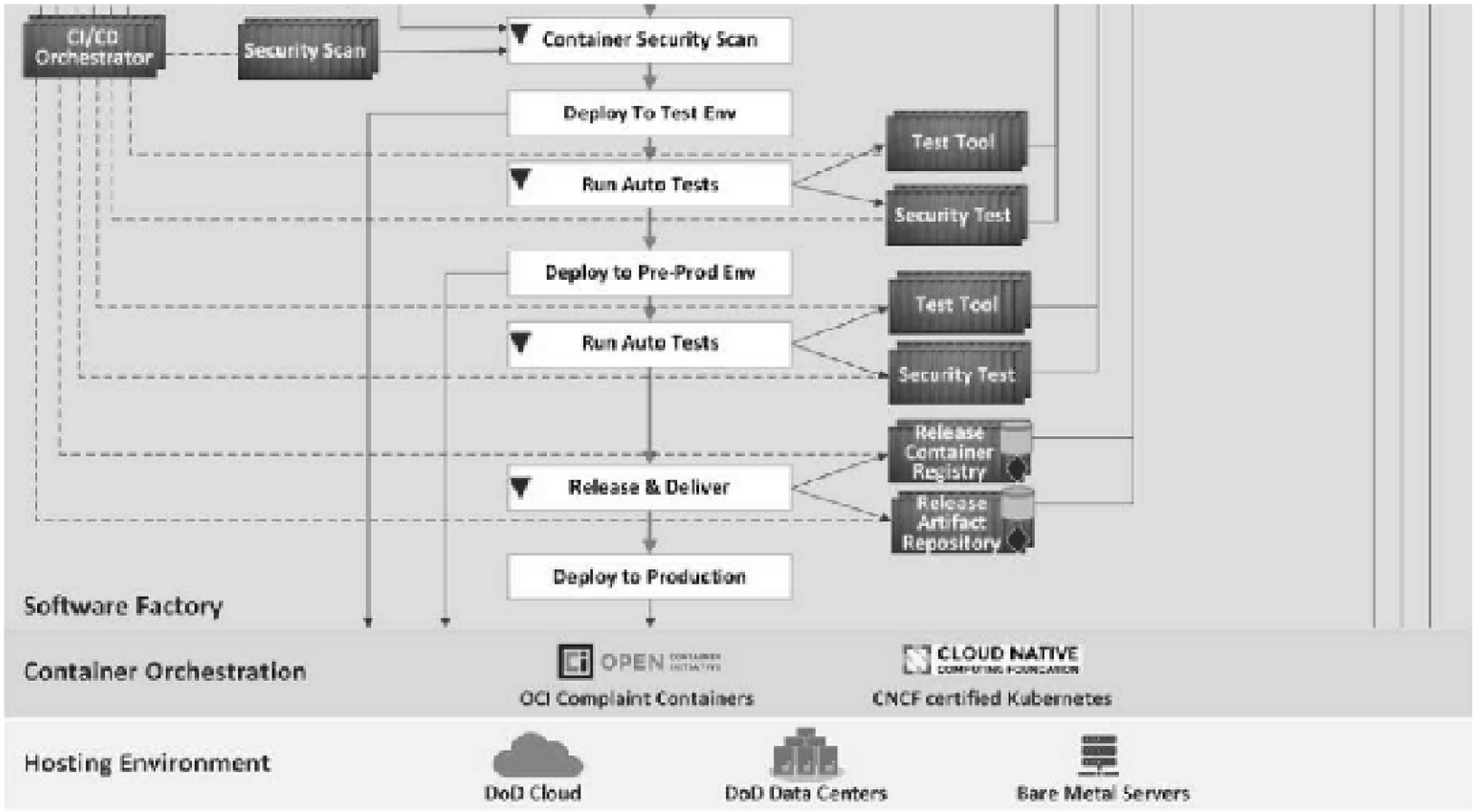


Fabrica


DoD Applications
Test Environment
Pre-production Environment
Production Environment
Tools
(Based on DoD Enterprise DevSecOps hardened containers)
Workflows
IDE
Code Repository
Build Tool
Security Test
Test Tool
Artifact Repository
Container Registry
Build App
Static App Security Test
Unit Test
Publish Artifacts
Build Container Image
Development Environment Test

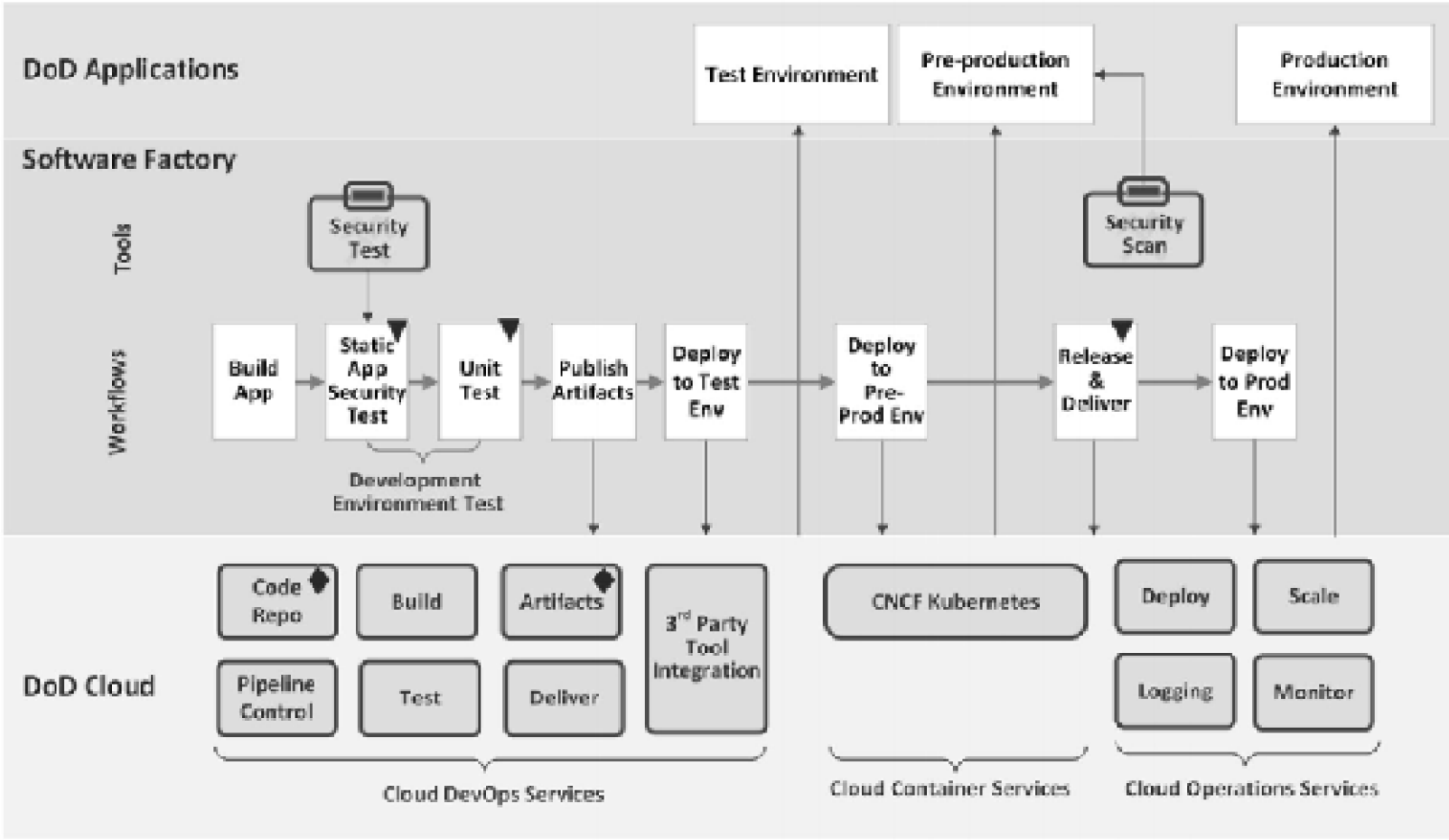
- X

Fabrica


CI/CD Orchestrator
Security Scan
Container Security Scan
Deploy To Test Env
Test Tool
Run Auto Tests
Security Test
Deploy to Pre-Prod Env
Test Tool
Run Auto Tests
Security Test
Release Container Registry
Release & Deliver
Release Artifact Repository
Deploy to Production
Software Factory
Container Orchestration
OPEN
OCI Complaint Containers
CLOUD NATIVE
CNCF certified Kubernetes
Hosting Environment
DoD Cloud
DoD Data Centers
Bare Metal Servers

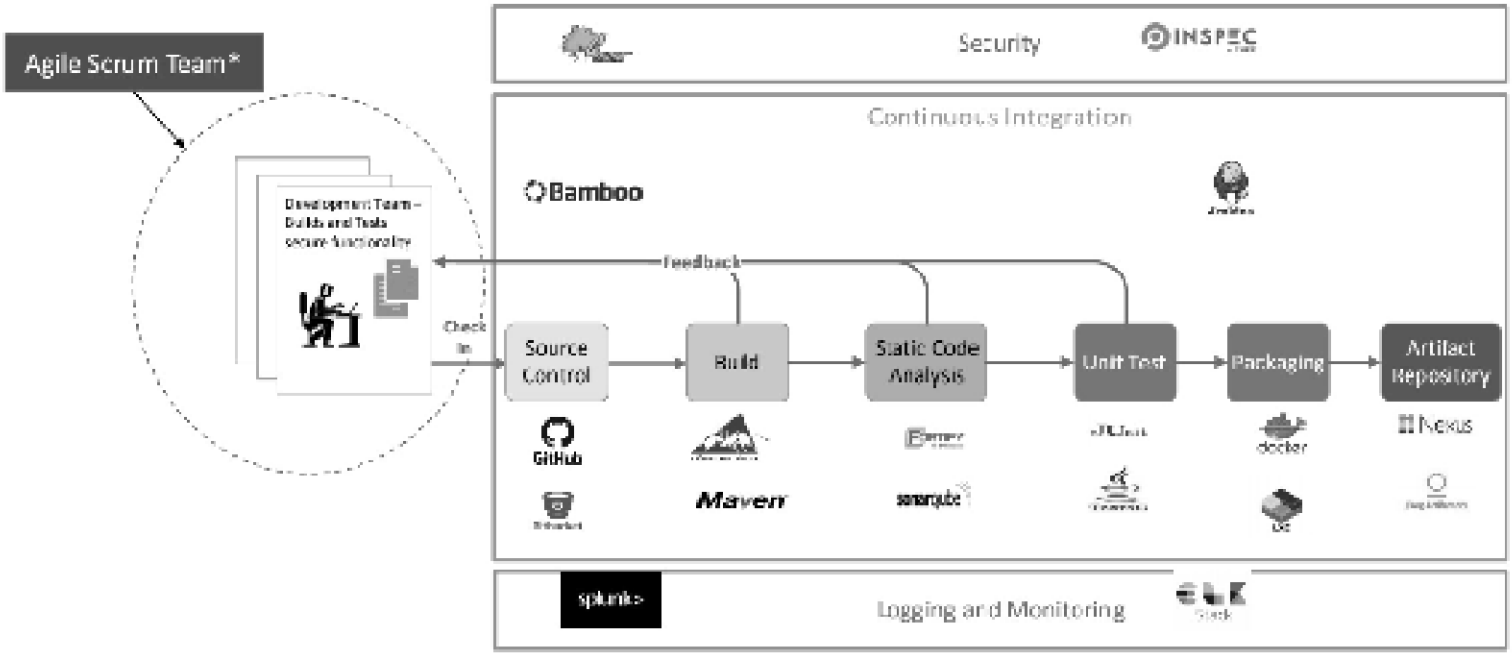
- x

Fabrica


DoD Applications
Test Environment
Pre-production Environment
Production Environment
Software Factory
Tools
Security Test
Security Scan
Workflows
Build App
Static App Security Test
Unit Test
Publish Artifacts
Deploy to Test Env
Deploy to Pre-Prod Env
Release & Deliver
Deploy to Prod Env
Development Environment Test
DoD Cloud
Code Repo
Build
Artifacts
3rd Party Tool Integration
CNCF Kubernetes
Deploy
Scale
Pipeline Control
Test
Deliver
Logging
Monitor
Cloud DevOps Services
Cloud Container Services
Cloud Operations Services

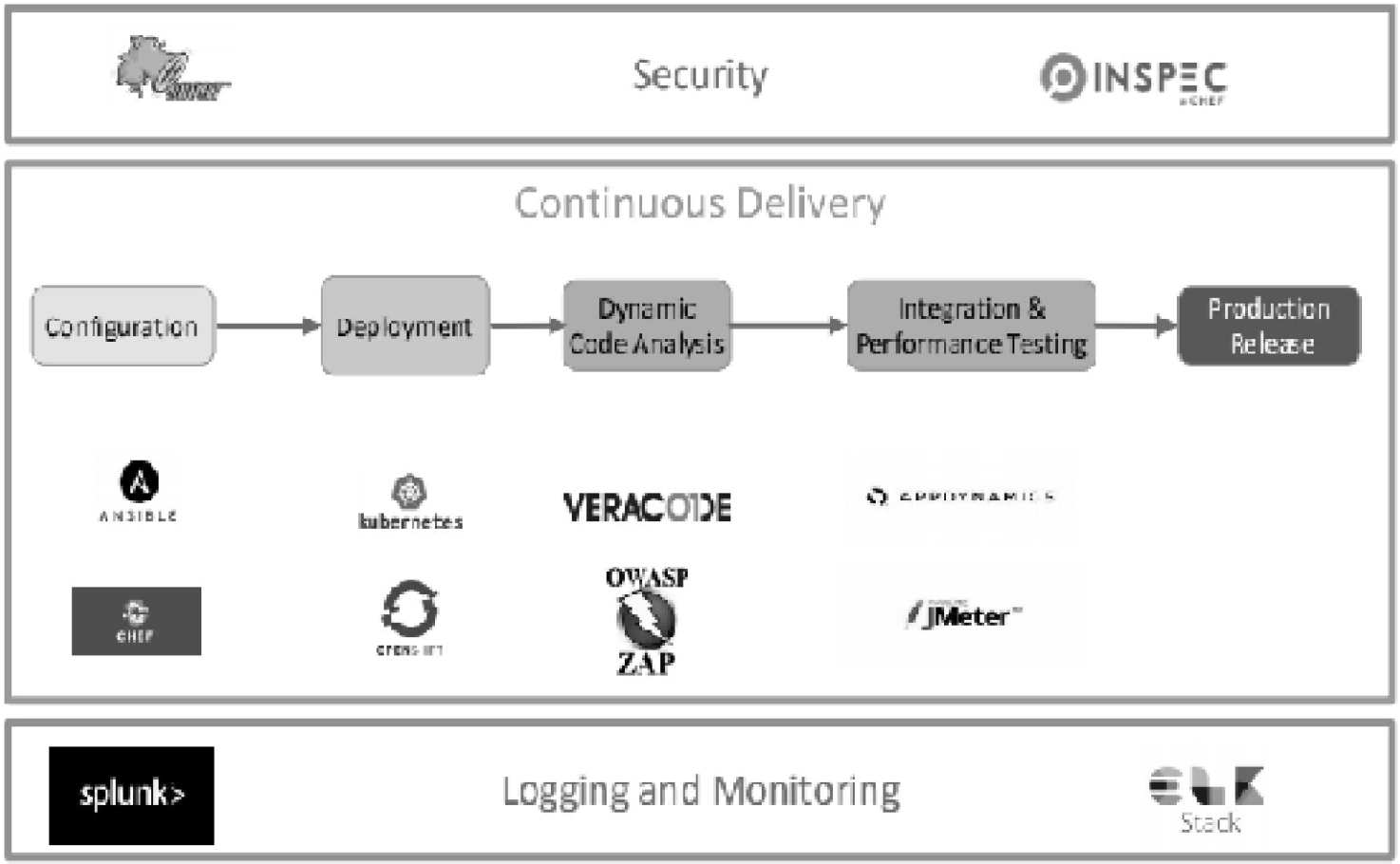
- X

Integrarea continua (CI)



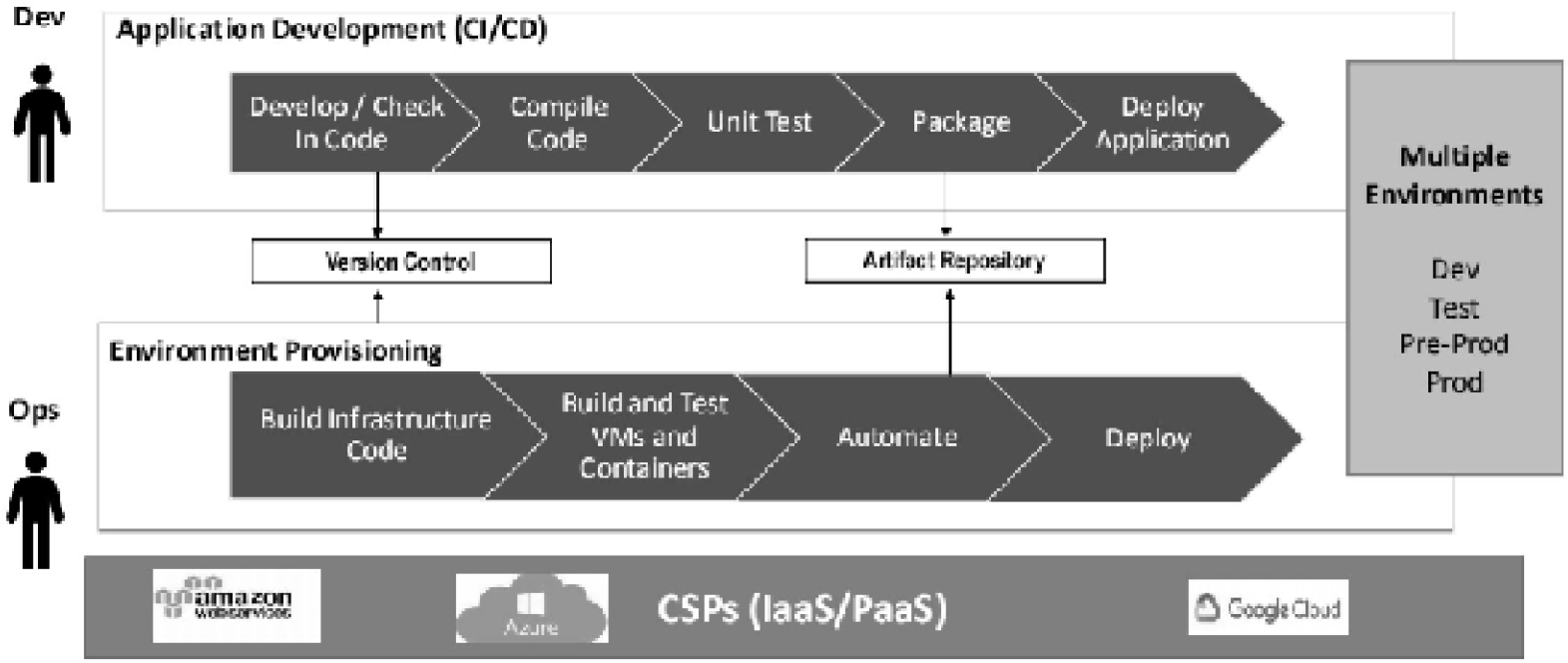
- X

Livrarea continua (CD)



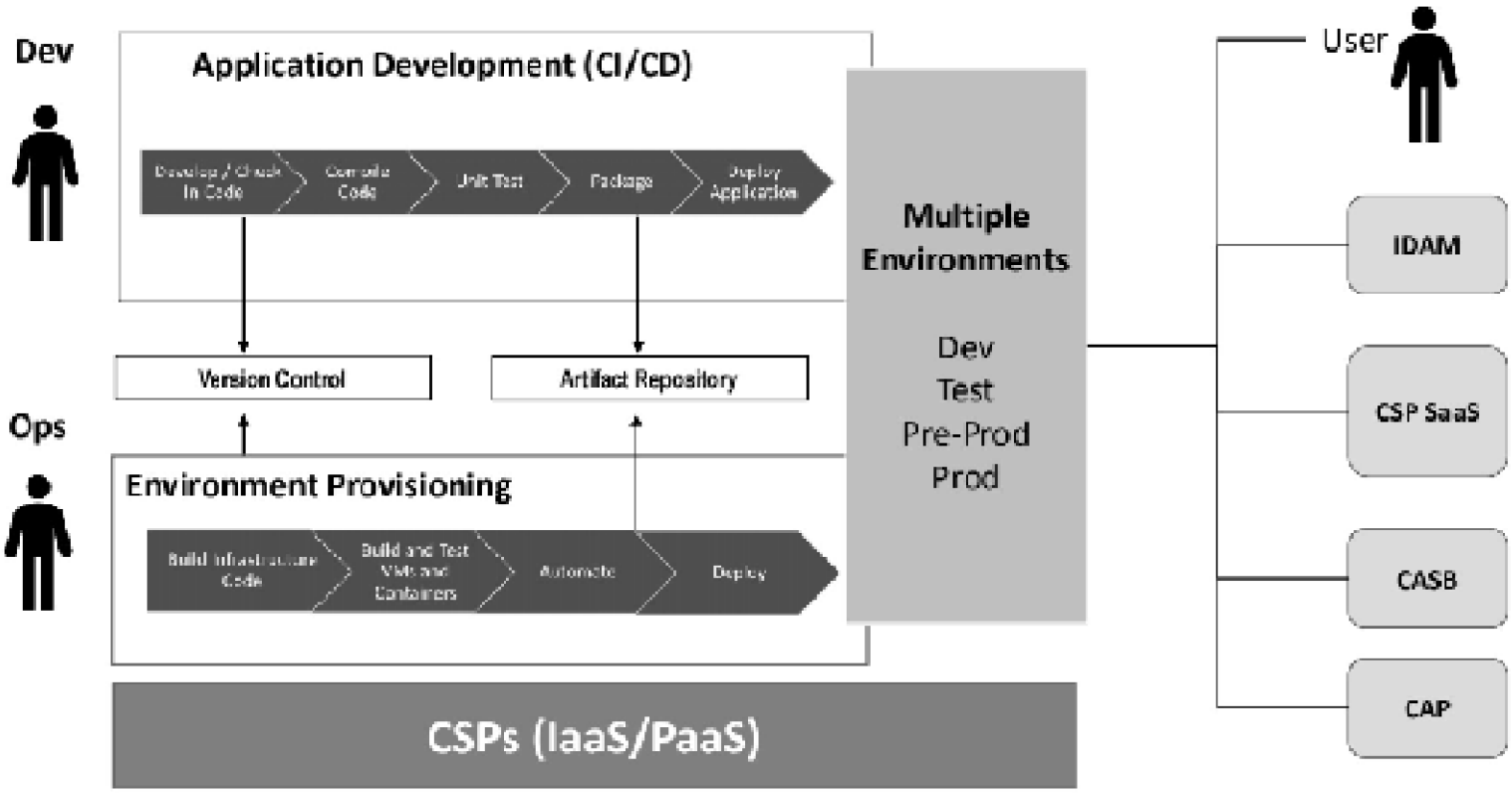
- X

CI - CD în nor



Cum se aplică în operații sistemul de integrare și livrare contiunuă (CI-CD) în nor

CI - CD în operații



- Cum se aplică în operații sistemul de integrare și livrare contiunuă (CI-CD) în operații

Securitatea în DevSecOps

- Analiza statică a codului
- Urmarirea și gestiunea configurației
- Analiza dinamică a codului
- Cautarea de vulnerabilități
- Infrastructura ca un Cod
- Monitorizare continuă
- Securitatea Containerului

Instrumente specifice asigurării securității în DevSecOps

Instrumentsecuritate	Descriere	UNDE	Gratuit
Snort		OT&E	DA
Fortify SCA		DT&E	Nu
Gauntlt		DT&E	DA
HashiCorpVault		DT&E	DA
Sonar Qube		DT&E	DA
OWASP Zap		DT&E și OT&E	DA

Cum intervine testarea în diverse etape ale lanțului CICD

- **Intrare**
  - metodologie
  - specificații
  - testare

- **Măsuri proactive**
  - La achizitii.
  - Entitatile necesare pentru testare
  - Testabilitatea

Cum intervine testarea în diverse etape ale lanțului CICD

- **Pre-Dezvoltare**
  - Architectura aplicatiei,
  - cazurilor de utilizare,
  - scenarii,
  - specificarea functionalităților
- **Măsuri proactive**
  - Interfete
    - Mediul de test
    - Datele de testare

Cum intervine testarea în diverse etape ale lanțului CICD

- **Dezvoltare**
  - Specificațiile de proiectare,
  - demonstarții,
  - testare
  - evenimente de testare
- **Măsuri proactive**
  - Noile interfete si surse de date
  - Trasabilitatea
  - Observatia
  - Evaluarea Riscului

Cum intervine testarea în diverse etape ale lanțului CICD

- **Ieșire**
  - Rapoarte de testare,
  - aplicația funcțională
  - modelele de date
- Măsuri proactive
  - Analiza ieșirii testelor
  - Recomandări

Instrumente specifice DevSecOps

- Pentru securitate
  - Snort, Splunk, Fortify SCA, Vault, OWASP Zap, SonarQube
- Pentru controlul codului sursă
  - GitHub, GitLab, Bitbucket, Artifactory
- Instrumente pentru integrare continună
  - Jenkins, Bamboo
- Instrumente pentru testare
  - JUnit, Selenium, JMeter, TestNG, SoapUI

Instrumente specifice DevSecOps

- Instrumente pentru configurare initiala/continua
  - Ansible, Chef, Puppet
- Instrumente de monitorizare si jurnalizarea executiei
  - ELK (Elasticsearch, Logstash & Kibana) Stack, Splunk
- Instrumente pentru orchestrare
  - Kubernetes, OpenShift
- Containere
  - Docker, Docker Swarm

Instrumente specifice DevSecOps

Instrument Testare	Descriere
JUnit	
Selenium	
SoapUI	
RationalFunctional Tester	
JMeter	
TestNG	
UnifiedFunctional Test (UFT)	

Abordări în testarea continuă

Testarea livrării continue

Descriere

Instalarea Albastru - Verde

Testul canarului

Testarea A/B