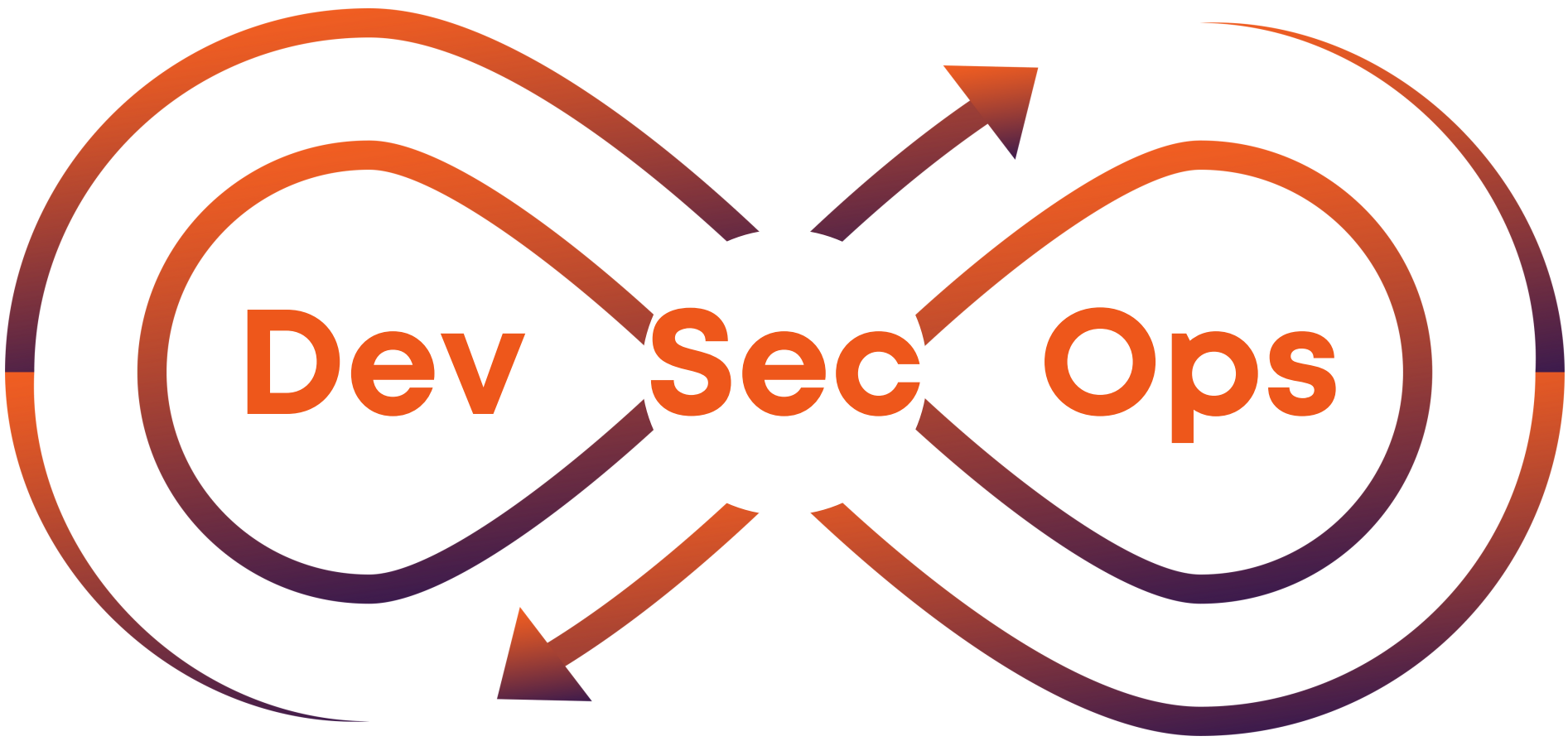
Demystifying

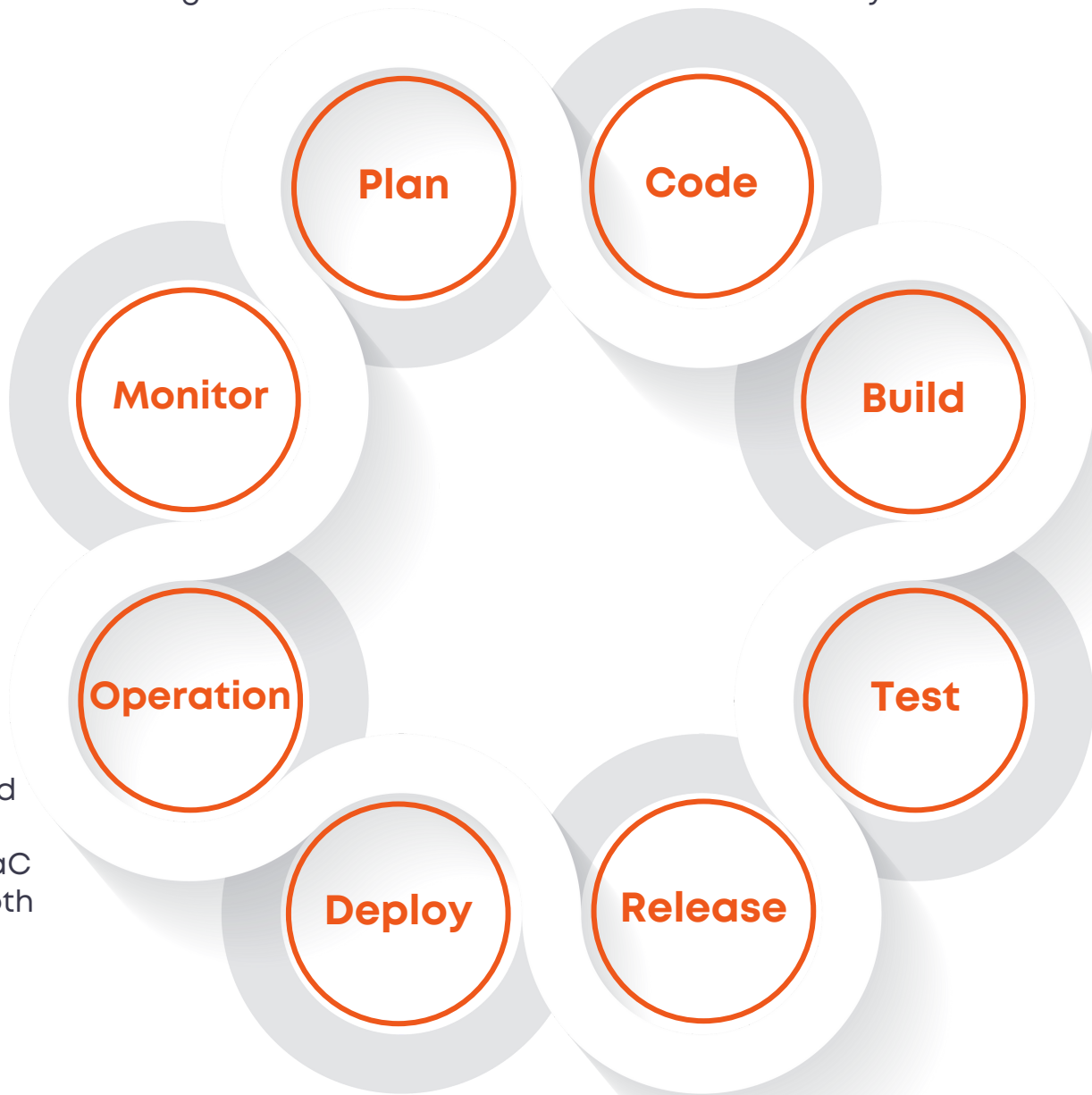


What are the steps involved in DevSecOps Pipeline?

Execute security analysis and create a plan for testing scenarios.

Use linting tools and Git controls to secure passwords and API keys.

To avoid a breach, use real-time monitoring tool to spot any abnormalities and exploits early on.



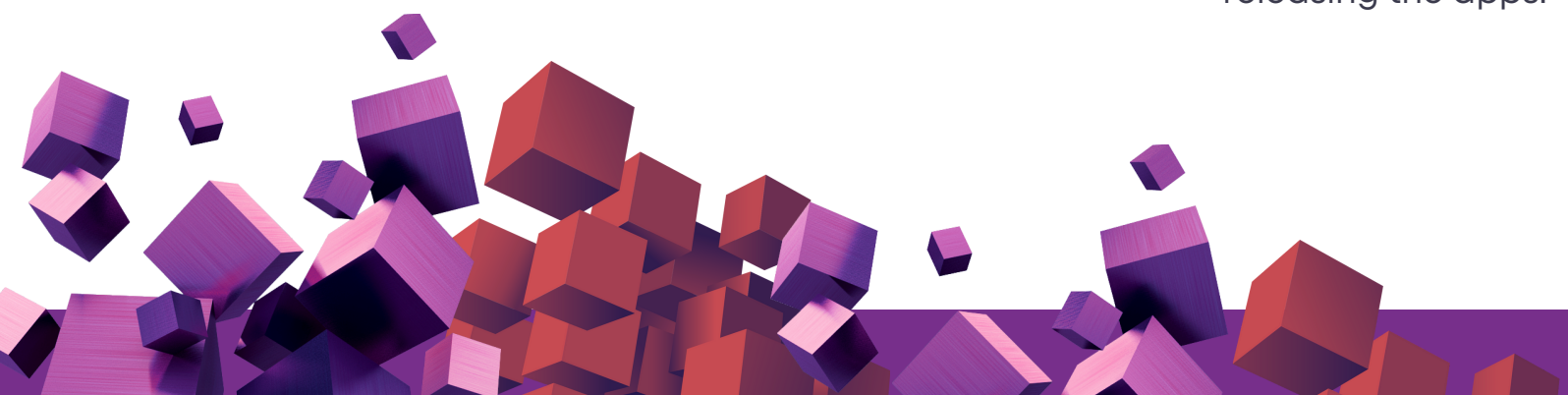
Use Static application testing (SAST) tools to track down flaws in code.

Periodic ops maintenance and DevSecOps integration with IaC tools ensure smooth operations and prevent vulnerabilities.

Identify user auth, SQL injection, API flaws with Dynamic App Security Testing tools. (DAST)

Send a secure infra or build to production for final deployment after completing the above tests in runtime.

Use security analysis tools for vulnerability scanning and penetration testing just before releasing the apps.



DevSecOps Checklist

PLAN

- Design Security Requirement
- Privacy Requirement
- Asset Mgmt. Requirement
- Architecture Security Requirement
- Secure Coding Best-practices Requirement
- Threat Modelling
- Compliance Requirement
- Infrastructure Security
- Requirement

CODE

- Credentials & Key Mgmt.
- SCA
- SAST
- Security Test Cases Generation
- Manual Code Review
- IDE Integrted Security Solutions & Plugins
- Source code Mgmt.
- IAM (Groups, Roles, POlicies & Permissions)
- Data-Backup, Restore, Recover
- Coding Platform Security & Best-practices (GitHub, GitLab, etc.)

BUILD

- Infrastructure as Code Security
- Data Security
- SSO
- Least Privileges
- IAM (Groups, Roles, Policies & Permissions)
- Service Accounts
- Dependency
- Database Configuration Security
- Application Configuration Mgmt.
- Build Automation
- Container Registry
- IAST
- Image Security
- Complaince
- Network Security
- AAA (AuthN/AuthZ/Accountability)
- Container Orchestration
- Cluster Mgmt.
- Endpoint Security
- Cloud Security POsture Mgmt.
- IAM (Groups, Roles, POlicies & Permissions)
- Microservice Security
- Content Delivery & Protection

INTERNAL SECURITY TESTING

- DAST
- Functional Security Testing
- Internal Red Teaming
- Vulnerability Mgmt.
- Security Scanning through Open-Source Tools
- Purple Teaming Exercise

RELEASE/ DEPLOYMENT

- Dockers & Kubernetes Security
- IAM (Group, Roles, Policies & Permissions)
- Build Process Automation Security & IAM
- UAT to Production Security
- Private Connectivity & Remote Access
- Security Detection & Monitoring Solutions Implementation Security & Review (WAF, SIEM, SOAR, DDOS, etc.
- Secure & monitor the entire physical and virtual environment
- Certificate Mgmt.
- Segregation of duties (process, hardware, software, environment)

EXTERNAL SECURITY TESTING

- Penetration Testing
- External Red Teaming
- Bugbounty

SECURITY MONITORING

- Security Operations Monitoring
- Security Event Monitoring
- Security Incident Mgmt. and Monitoring
- Threat Intelligence Exercises
- Threat Hunting Exercises
- SIEM
- SOAR
- WAF Alerts
- DDOS Detection & Protection
- Packet Analysis
- IDS/IPS Implementation Security & Review

RUN/ OPERATE

- Bug Fixes
- Patch Mgmt.
- Operational Security
- Software Release Integrity Check Security
- New Release Security & Automation
- Incident Mgmt. & Response
- Cross-team Security Awareness Training (Building Culture)
- Version Control & Metadata Mgmt.
- Compliance Mgmt.
- Logging, Auditing & Monitoring

Examples:

DevSecOps

Tools

- IDE Plugins
- Pre-Commit Hooks
- Secrets mgmt. tools
- Source Composition Analysis (SCA)
- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Interactive Application Security Testing (IAST)
- Secure infrastructure as code

- Compliance as code
- Runtime application self-protection (RASP)
- Web Application Firewall (WAF)
- Monitoring tools
- Chaos engineering
- Vulnerability management
- Manage vulnerability risks
- Container security tools
- CI/CD tools
- Secure coding tools
- Security policy mgmt. tools

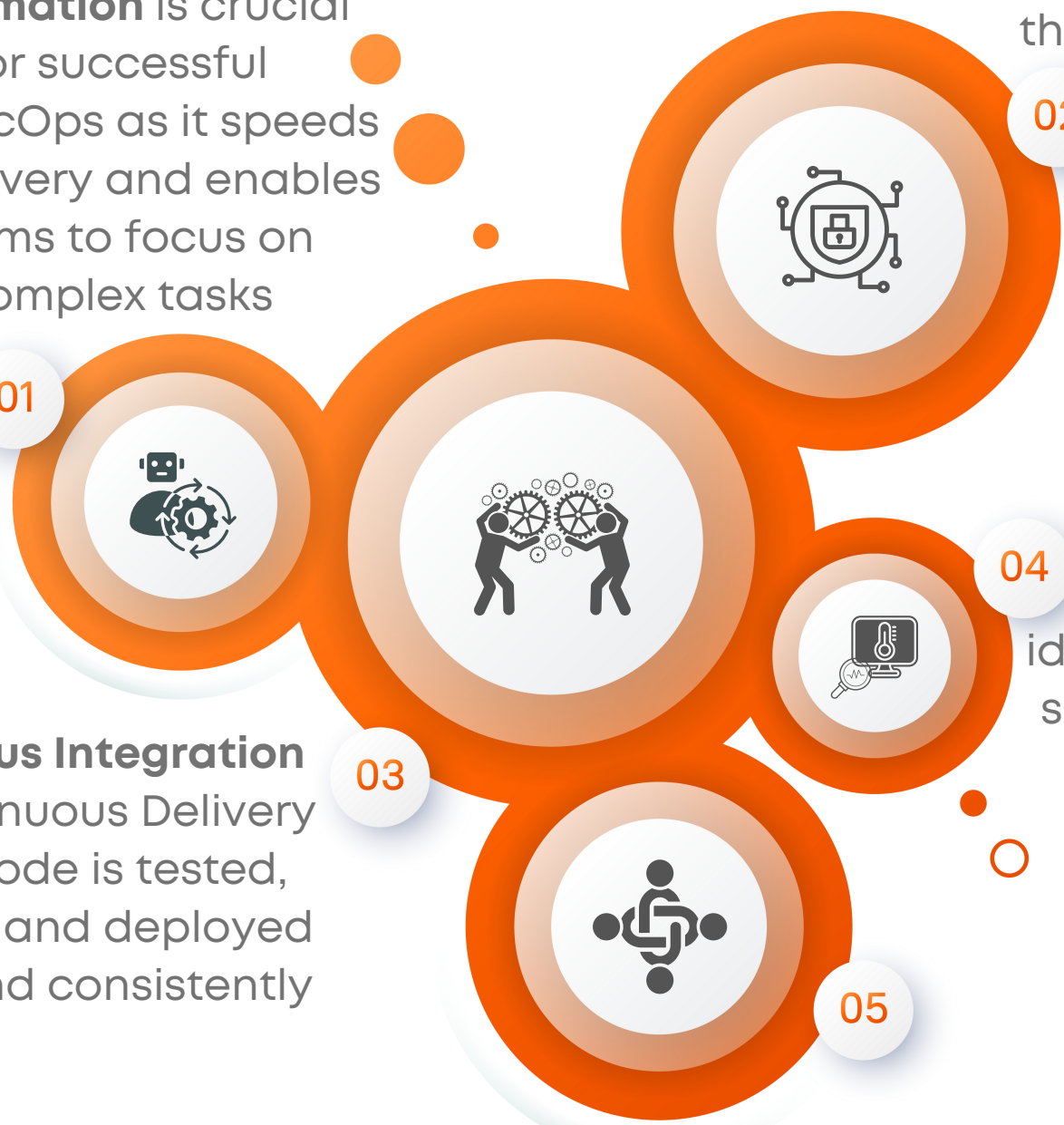
What are the principles of DevSecOps Pipeline?

Automation is crucial for successful DevSecOps as it speeds up delivery and enables teams to focus on complex tasks

01

Shifting Security left reduces risks by integrating security practices earlier in the development process

02



Continuous Integration and Continuous Delivery ensure code is tested, reviewed, and deployed quickly and consistently

03

Measuring and monitoring are essential for identifying potential security issues and addressing them quickly

04

Collaboration and communication are key to successful DevSecOps as teams need to share ideas and work together to ensure proper security implementation

05

What problems does DevSecOps Solve?

DevSecOps incorporates security into every stage of the development process, reducing the potential for costly security incidents.

DevSecOps provides greater visibility into the security of applications and helps identify potential risks much earlier in the development cycle.

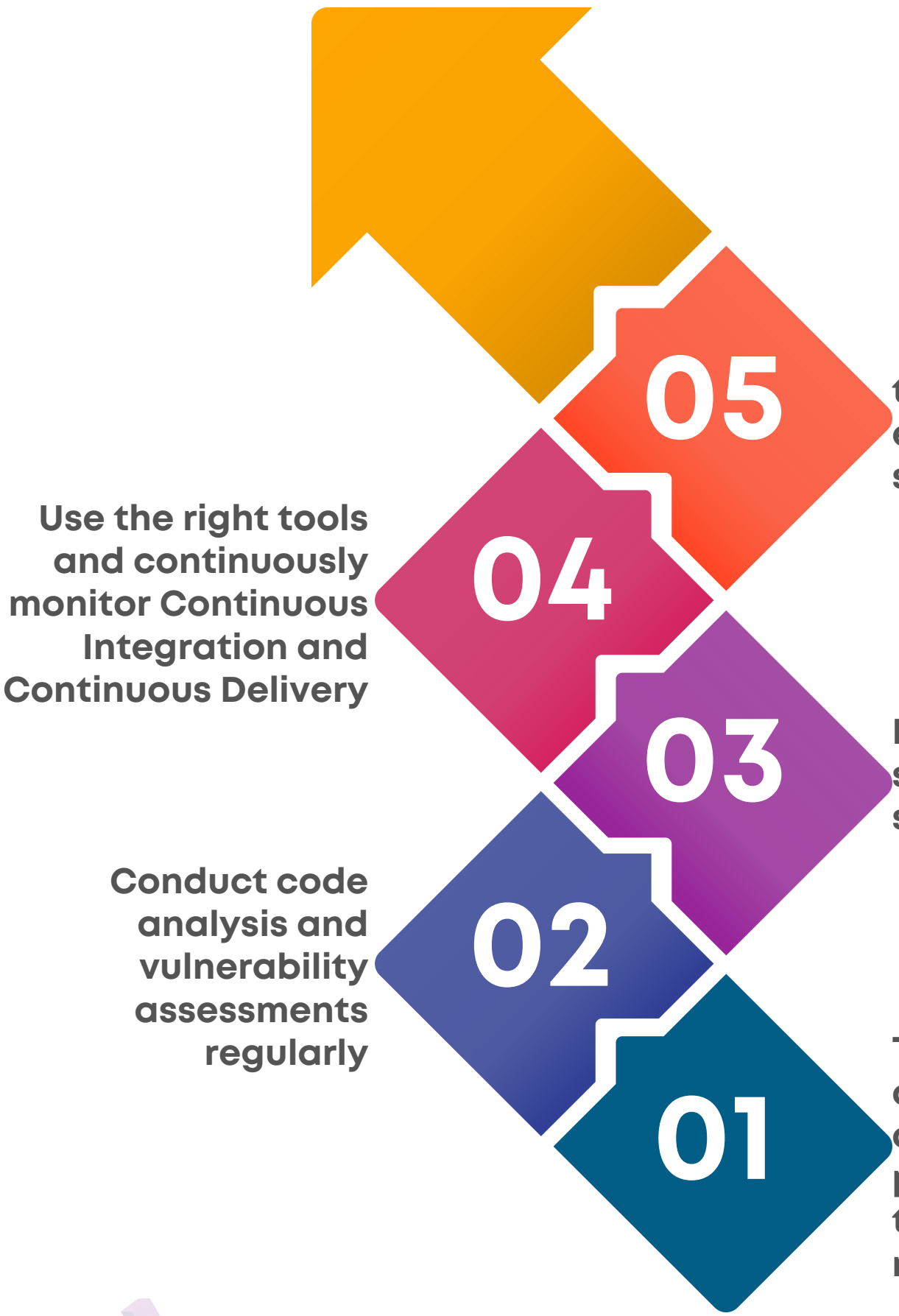
DevSecOps streamlines development processes by automating tasks and eliminating manual steps, resulting in faster and more reliable delivery of software.

DevSecOps tools can automate the development, testing, deployment and maintenance of applications while ensuring security and compliance requirements are met.

DevSecOps encourages collaboration between development and security teams to identify and address security issues, leading to improved communication and a more secure development process.

Effective ways to adopt

DevSecOps



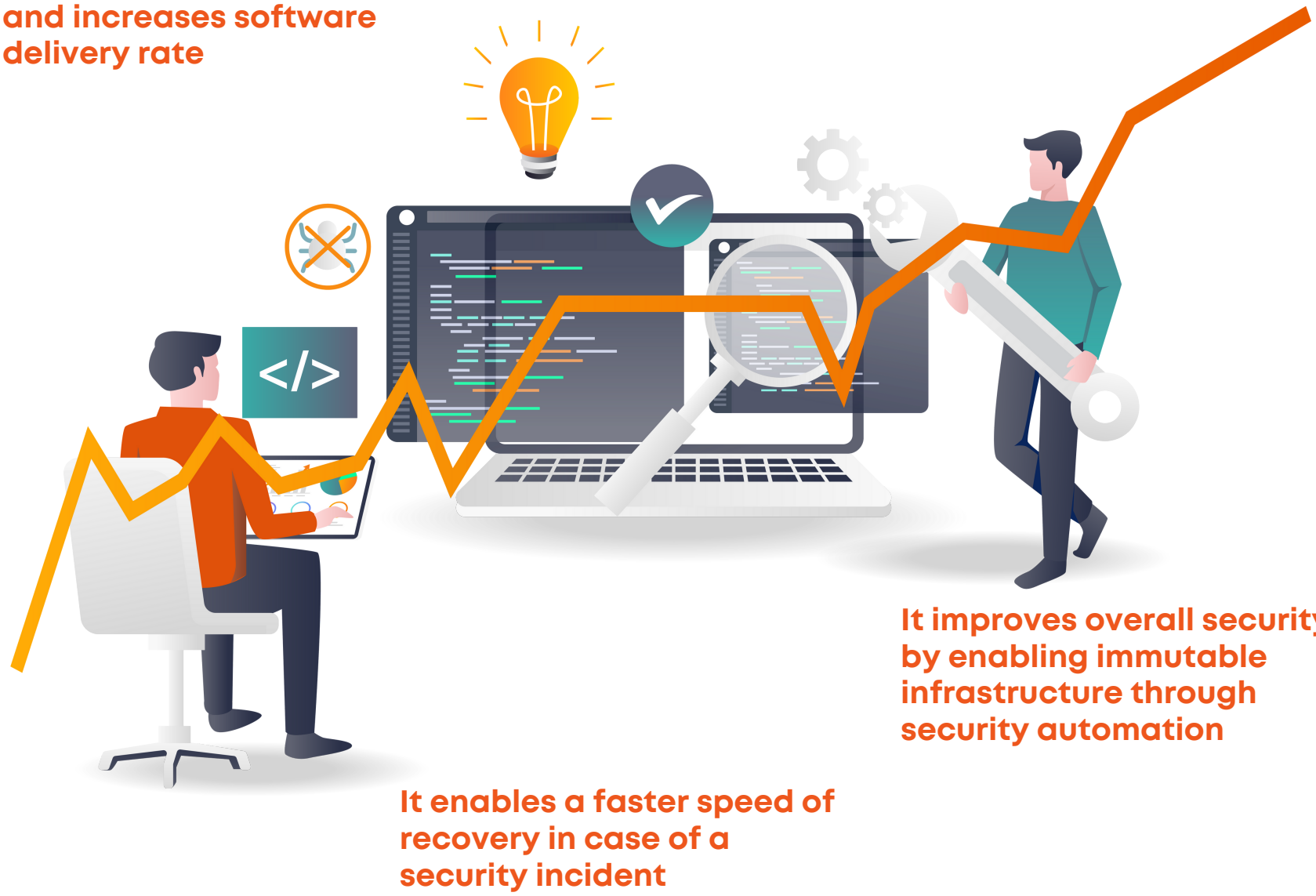
Following these steps can lead to improved security, reduced risk, and increased efficiency

Benefits of DevSecOps

It provides enhanced transparency throughout the development process

This approach enables security checks, continuous monitoring, and automated deployment checks from the beginning of application development

It reduces compliance costs, speeds up application deployment, and increases software delivery rate



It improves overall security by enabling immutable infrastructure through security automation

It enables a faster speed of recovery in case of a security incident

The Secure by Design principle and the ability to measure security aspects are embedded in this approach

Best-practices of DevSecOps

- Maintain security throughout the software development process
- Train and adopt secure coding practices
- We need to select the appropriate processes and integrate them into the DevOps pipeline.
- Choose appropriate tools for security checks
- Implement security scanning tools to detect vulnerabilities
- Move to Git as a single source of truth
- Know code dependencies
- Use an analytics-driven SIEM platform
- Use container security solutions to secure containers
- Implement Continuous Integration/Continuous Delivery (CI/CD)

- Set up security policies and enforce them throughout the organization
- Use software composition analysis to identify open source components.
- Discover vulnerabilities first and pinpoint their exact location
- Enforce licensing policies automatically at scale
- Prevent known and unknown OSS risk from entering the SDLC

Article by

Praveen Singh

Infographics by

Netpoleon India

Techtalk Series-An initiative by

Mohan Kumar T L