To The Left - How Beyoncé Can Help Us Develop and Deploy Secure Code

Aditi Chaudhry

@aditichaudhry92

Overview

- What is DevSecOps?
- Why do we need it?
  - How is it different than DevOps?
- What benefits does it provide?
- What challenges does it face?
- Implementation Suggestions
- Questions?

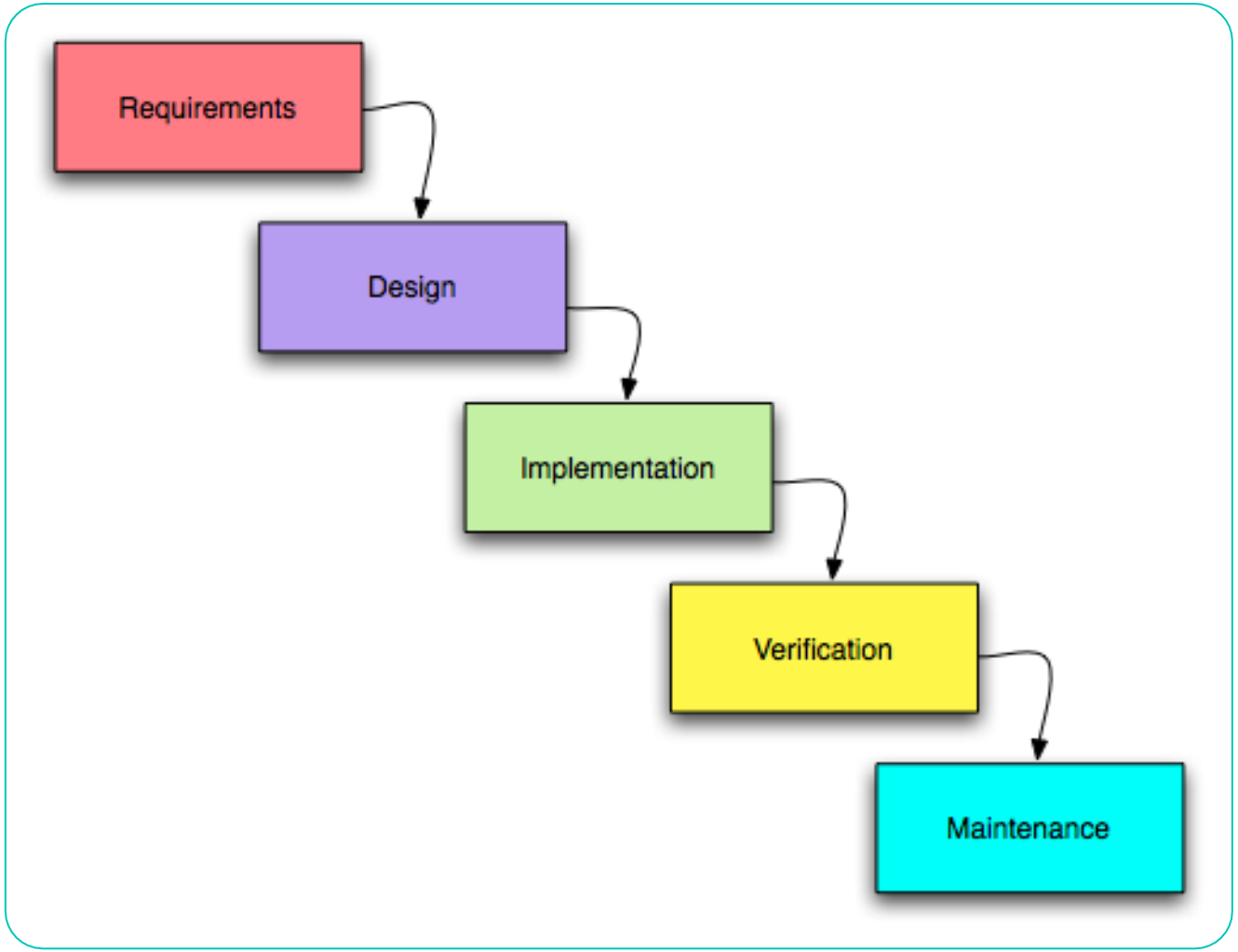
Intro

What is DevSecOps?

- Introduce security early in SDLC
- Everyone involved in security

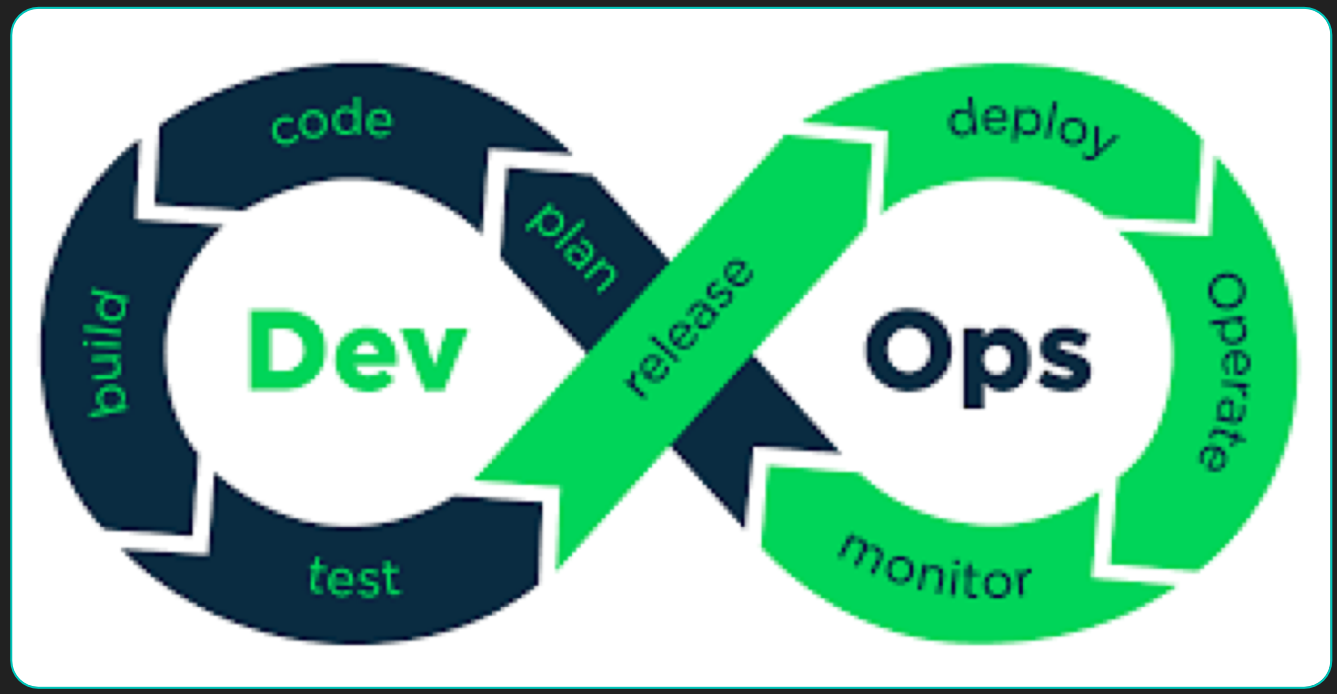
Why We Need DevSecOps

- Software products followed the waterfall methodology
  - Linear sequential approach for developing a product
- Concluded with a "big-bang" release



DevOps

- Strongly advocates for automation & monitoring at all steps of SDLC
- Goals:
  - Shorter development cycles
  - Increased deployment frequency
  - More dependable releases
  - All aligned with business objectives



Security Challenges in DevOps

DEVOPS

stable infrastructure/applications !=

secure infrastructure/applications

Security Challenges in DevOps

- Security was seen as the last gate-check on the way to a production release

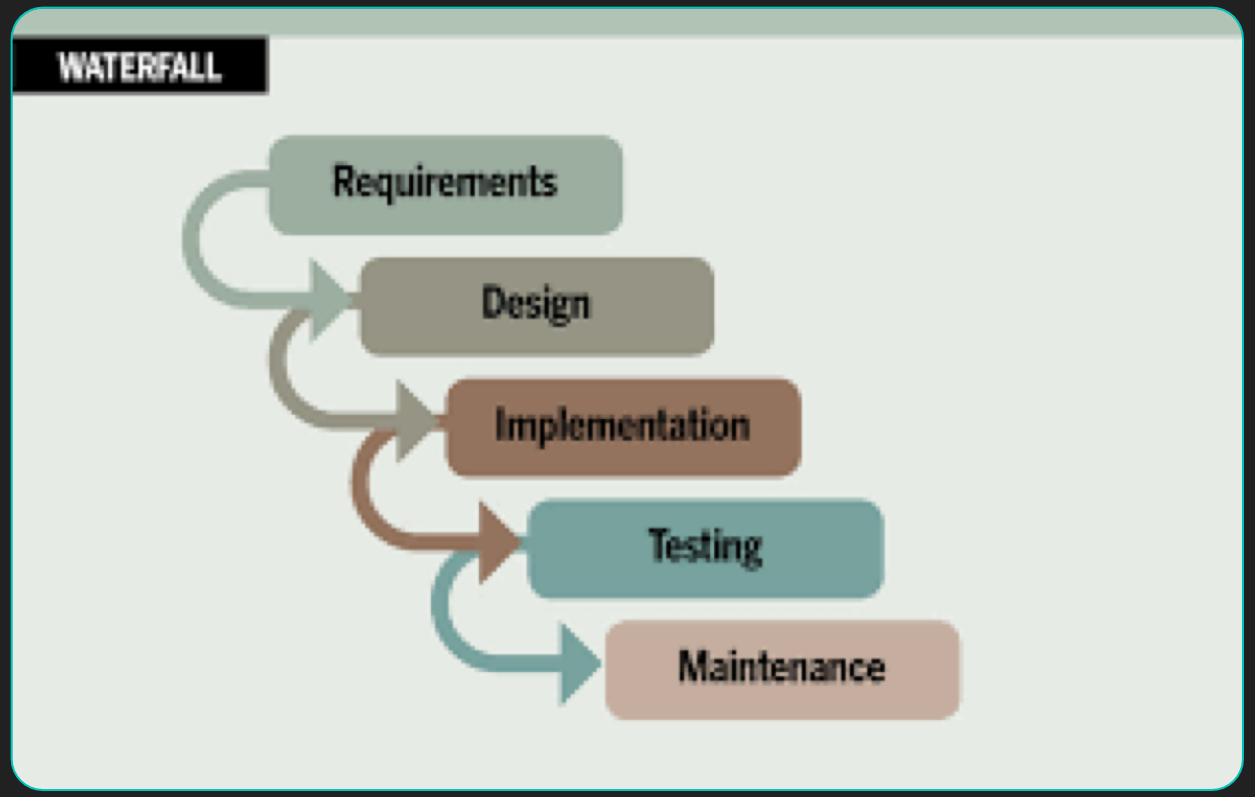


Figure 1. Information Security Professionals: Do You Believe Your Information Security Policies/Teams Are Slowing IT Down?

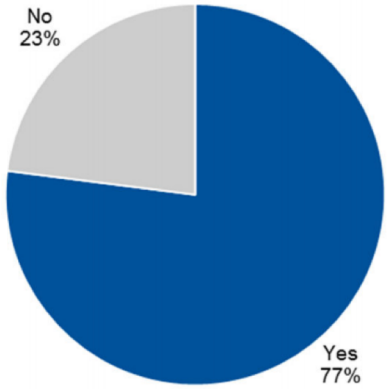
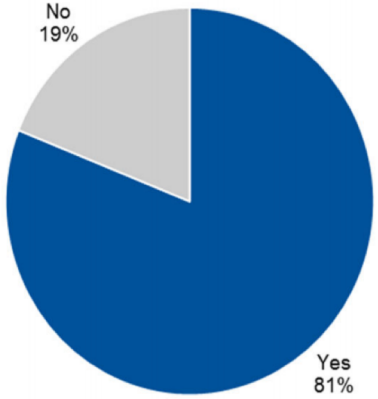


Figure 2. IT Operations Professionals: Do You Believe Your Information Security Policies/Teams Are Slowing IT Down?

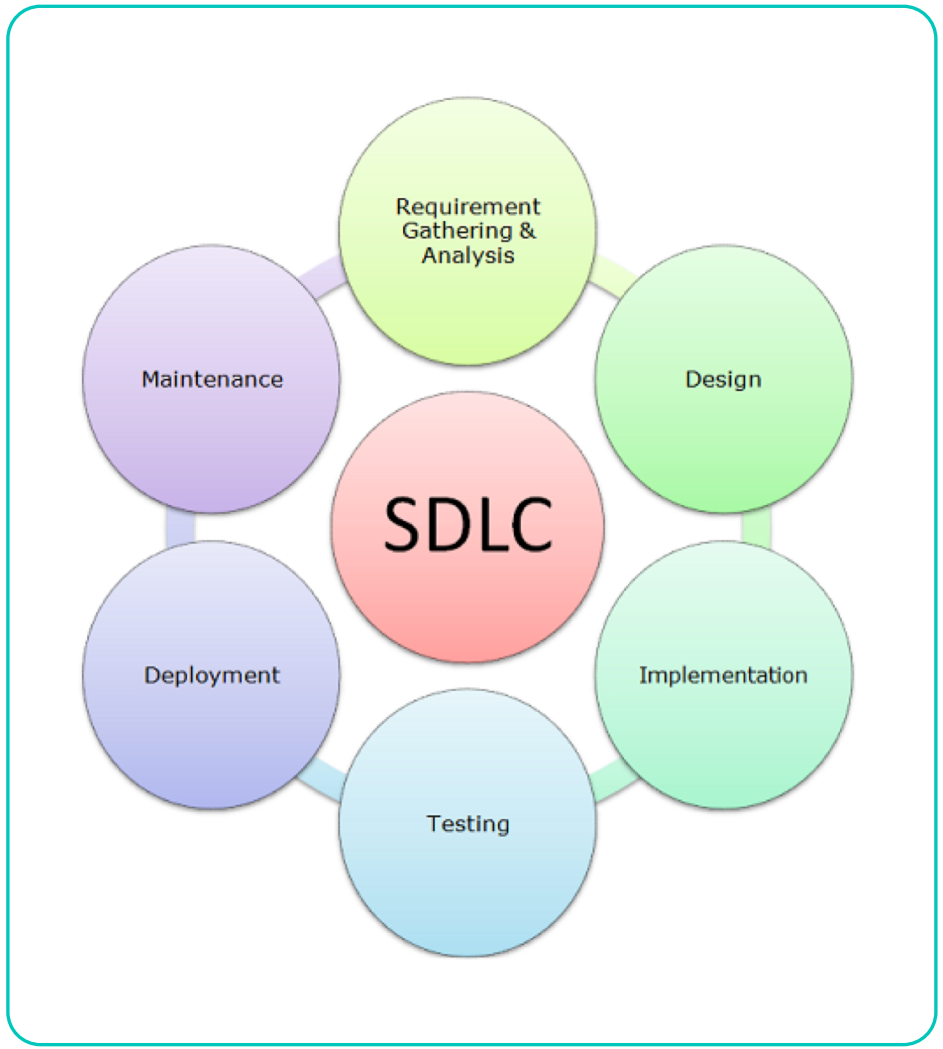


Security Challenges in DevOps

- Concerning that both InfoSec and IT ops professionals believe InfoSec is slowing down tech's ability to respond to the needs of the business

Security Challenges in Agile

- Focus is rapid delivery
- Teams can continuously align product deliverables to business needs through iterative planning & continuous feedback
- If you're releasing a new version of your product every week, when do you test for security vulnerabilities?



Security Challenges in Agile

Traditional security processes have not kept pace in agile/DevOps environments

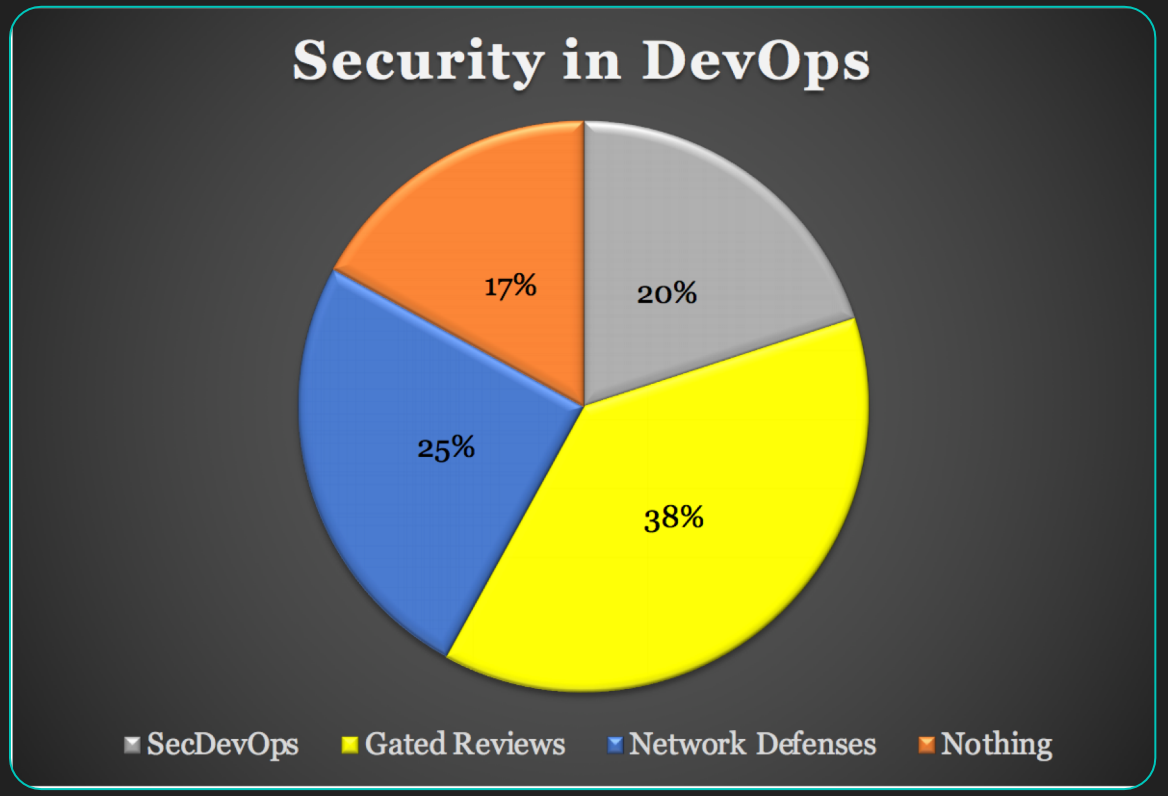
DevOps has been viewed as a risk by security teams

Increased velocity of software releases are seen as a threat to governance, security & regulatory controls

Result is that security becomes a major roadblock in software development where it is usually bypassed

Current State of DevOps Security

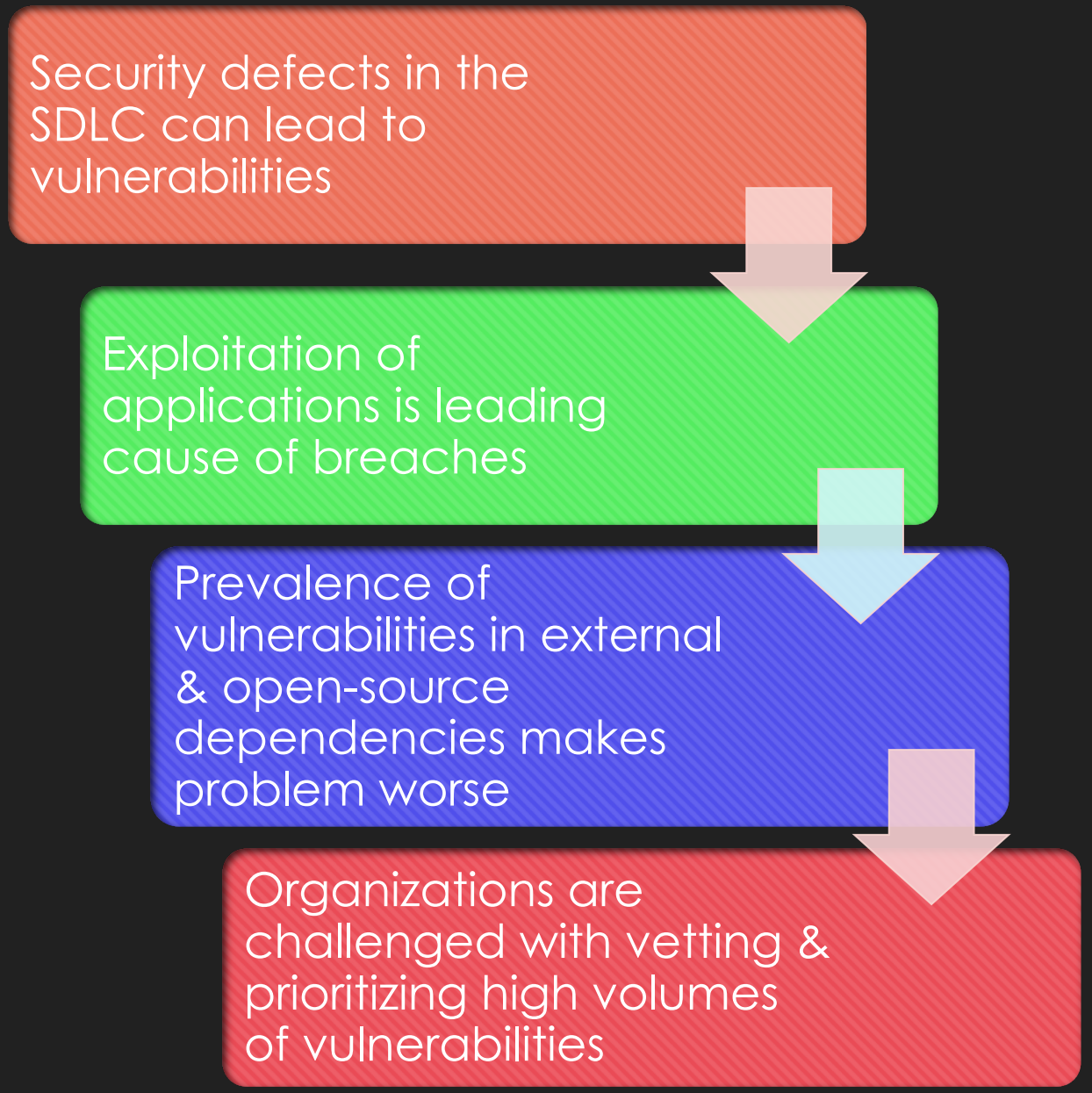
- A HPE study in 2016 found that security is being-shortchanged on DevOps teams
  - 20% do security in development/delivery
  - 38% still depend on pen-testing or other pre-production gate reviews
  - 25% rely on network defenses
  - 17% do nothing for security
- Overall sentiment is that security is someone else's problem



Application Risks

- If security is not bypassed, dev team rarely has enough time to address all issues before production deployment
- Insecure application lives somewhere on the internet
- Irony of ignoring security to avoid the risk of missing a deadline actually puts more risk into the application

Application Risks



Why We Need DevSecOps

Running application security testing (AST) tools manually or in the final stages of the SDLC is expensive & disruptive

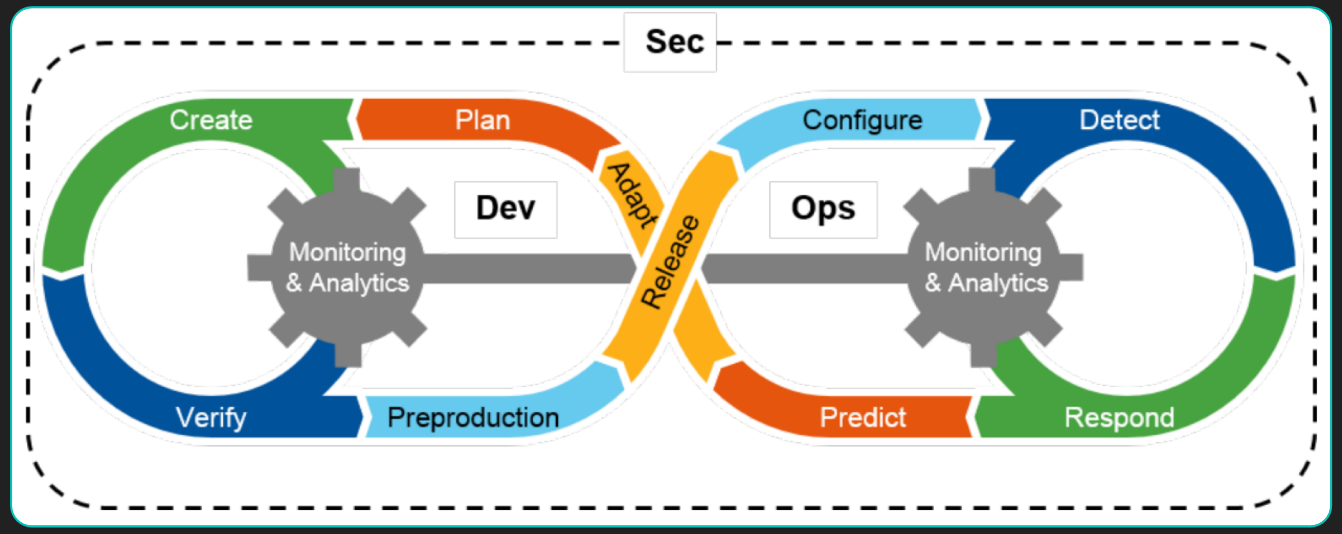
Software build pipelines have accelerated with the adoption CI/CD, necessitating integration and/or automation of AST

#goals

- Automatically incorporate security controls without manual configuration in a transparent way to DevOps teams & doesn't impede DevOps agility
- Fulfills legal & regulatory compliance requirements as well as manages risk

What is DevSecOps

- Where security checks & controls are applied automatically & transparently in rapid-development DevOps environments
- Most effective DevSecOps programs start at the earliest points in the development process



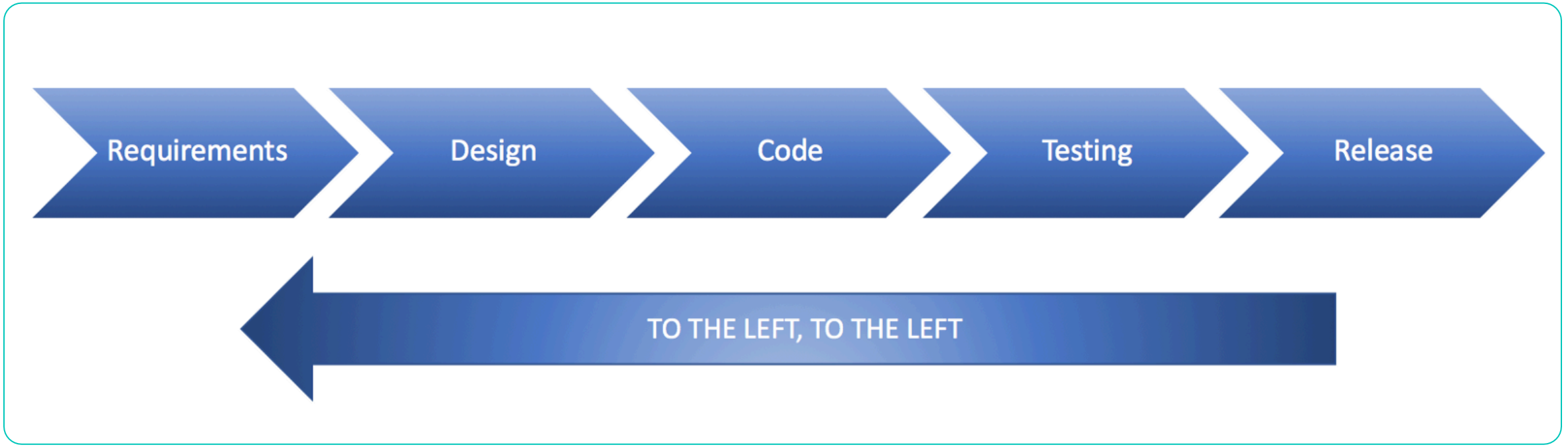
Goal of DevSecOps

Two seemingly opposing goals, “speed of delivery” & “secure code” are merged into 1 streamlined process

Intent is that “everyone is responsible for security”

Push security left and automating core security tasks

Beyonce and Security



To the Left

- By shifting left, teams can quickly discover & analyze vulnerabilities
- Can adapt code to mitigate against those vulnerabilities

Benefits

- Allows developers to focus on writing high quality & secure code
- Enables teams to release **titanium** applications

Benefits

- Security from the start minimizes chance of vulnerabilities
- Focus on high-hanging fruit
- Better collaboration & communication between dev & security teams
- Improved operational efficiencies across security & the enterprise
- Reduces chance of misadministration & mistakes

Challenges

Still ways to circumvent security checkpoints

Example 1

Vulnerability scanner is being used to block a build

Know what vulnerability is but you really need to do this release

Find a way to hide pieces of the code that you know will fail a security scan, resulting in a successful build

Example 2

Teams decide to break build if there's a presence of 1+ findings of a certain severity

Ex: team says that build only breaks when the finding is high or critical

Helps identify & address high priority issues immediately

Consequence is that medium and low findings make it to production builds

Implementation Suggestions

How do we practically integrate security into DevOps?

Implementation

- Traditional DevOps tools such as Jenkins and Git are a must have to build the foundation of DevOps pipeline
- Many security tools in the marketplace, range from open-source to proprietary solutions
- Can be integrated into existing pipelines

DevSecOps - Planning

Threat Modeling

- Start w/high-level risk assessment for new systems/services
- ex: STRIDE model

Access Control Ownership

- Provisioning
- Profiles
- Roles
- Identification

Resiliency

- Care about resiliency because it ties into the CIA triad model, specifically for availability

DevSecOps - Commit

Static Code Analysis

- ex: Checkmarx:
  A SAST Tool that analyzes an application's code for flaws which are indicative of security vulnerabilities

Open Source Analysis

- ex: WhiteSource:
  An open source vulnerability scanner, which runs automatically & continuously in the background

DevSecOps – Commit (2)

Unit/Integration Test Security

- Data security
- Protect against malicious code
- Unit testing frameworks exist for basically every language
- Security unit test ex: testing valid and invalid input
- Security integration test ex: testing unprivileged user access

Code review

- ex: Sonar:
  An open source platform to perform automatic reviews w/ static code analysis to detect bugs, code smells & security vulnerabilities on 25+ programming languages
- Peer/Team Code Review

DevSecOps - Deploy

- Logging of changes
- Secrets management
- Cloud/ infrastructure security
- Segregation of duties
  - Approver != implementer

DevSecOps - Operations

Prod Security Tests

- ex: Zaproxy - The OWASP Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications

Audit Logging

- ex: AWS uses CloudTrail to provide audit logging on different services it provides

Threat Intelligence

- Forecasts
- Alerting
- Reporting
- Resolution

Conclusion

- DevSecOps can create an effective and viable security layer for applications & environments
- It can serve as a solid foundation to ensure security & compliance in the long run, in a more streamlined, efficient, & proactive way

Questions?

Contact

@aditichaudhry92

linkedin.com/in/aditi-chaudhry-47303a80

medium.com/@aditi.chaudhry92