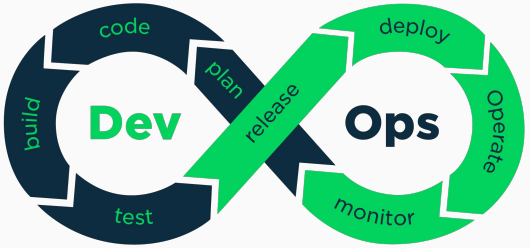
DevSecOps

Satya Ram Twanabasu

Why DevOps?

- Paradigm shift in software development
  - End of waterfall and emergence of Agile
- Rise of Containers
- Microservices
- Cloud technologies



What is DevOps?

- “DevOps is a set of software development **practices** that combines software development (Dev) and information technology operations (Ops) to **shorten the systems development life cycle** while **delivering features, fixes, and updates frequently** in close alignment with business objectives.”

  - Wikipedia entry of DevOps

Five aspects of DevOps

- Agile methodology
  - Small release cycles, faster adaptation to changes, frequent and quicker delivery
- Container technology with DockerFile
  - End of “It works in my machine”
- Automation
  - To give feedback without disturbing development, CD/CI
- Everything is code
  - Not only programming codes, but also config
- Communication and Collaboration
  - Continuous learning, feedbacks and suggestions

DevSecOps in Google Trend

DevSecOps
Search term

+ Compare

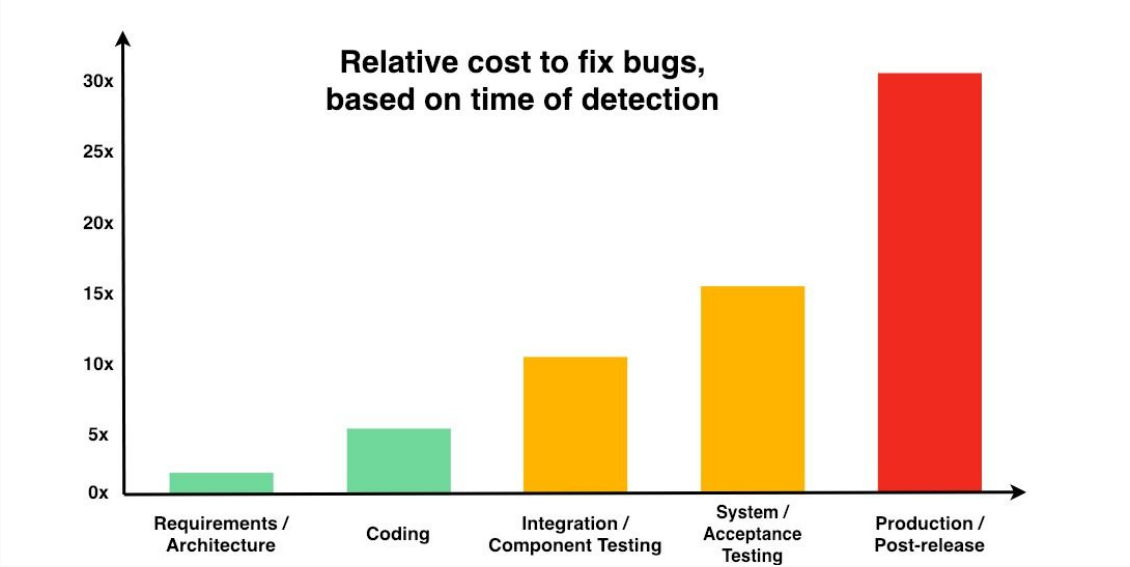
Worldwide | Past 5 years | All categories | Web Search

Interest over time

Background for DevSecOps?

- Security has been an after-thought.
- When coding, mostly the focus is only in getting things done but *writing codes that work is just a beginning*.
- When the security in focus in all the phase of development, right from the beginning, it is DevSecOps

Cost of fixing bugs



https://deepsource.io/blog/exponential-cost-of-fixing-bugs/

Why DevSecOps

- Security has different faces.
- Automates the integration of security at every phase of the SDLC
- In the past, security was 'tacked on' at the end of the development cycle by a separate security team
- Security as shared responsibility.
- Advantage : it decreases remediation time while making the product safer, lowering costs in the long run.

What is 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

https://www.devsecops.org/blog/2015/2/15/what-is-devsecops

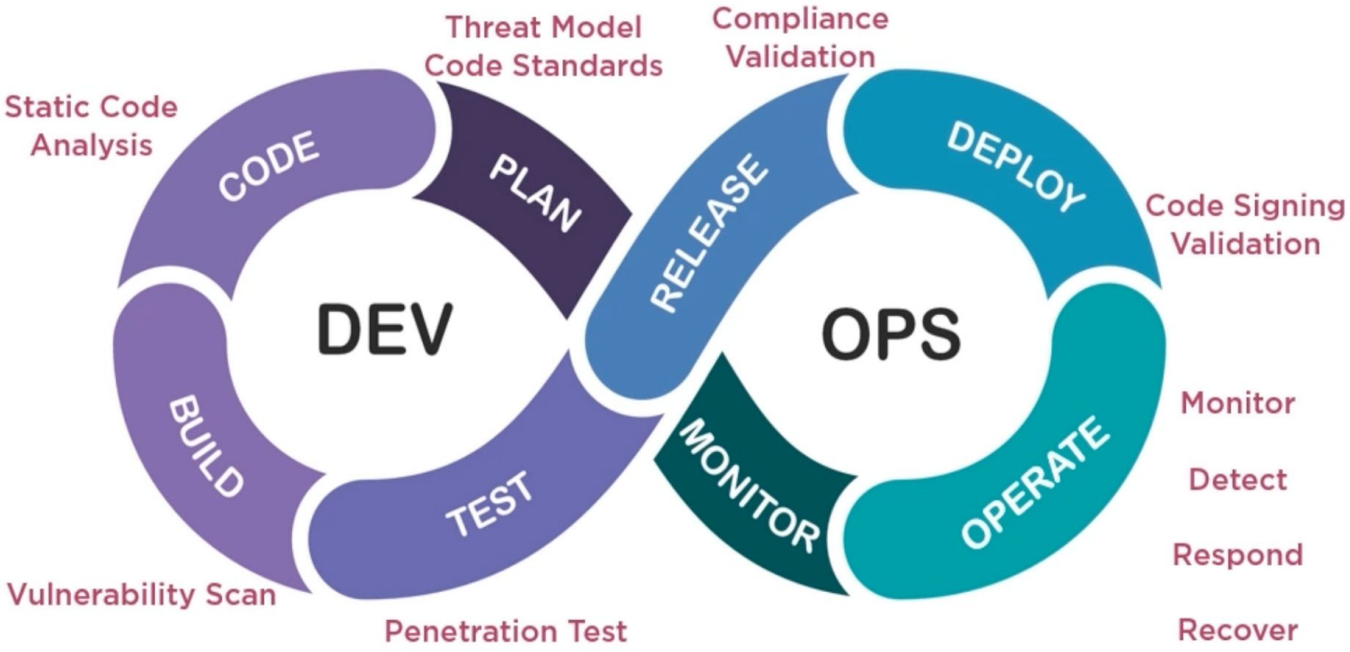
DevSecOps as Culture

“DevSecOps is a culture and a ***culture cannot be forced***; ***it has to be nurtured***. Strong development culture is based on **candid communication** between team members, **respect** people have for one another, and the celebration of technical artistry. ***Together***, ***culture flourishes***.”

- DSO Community Survey, 2020

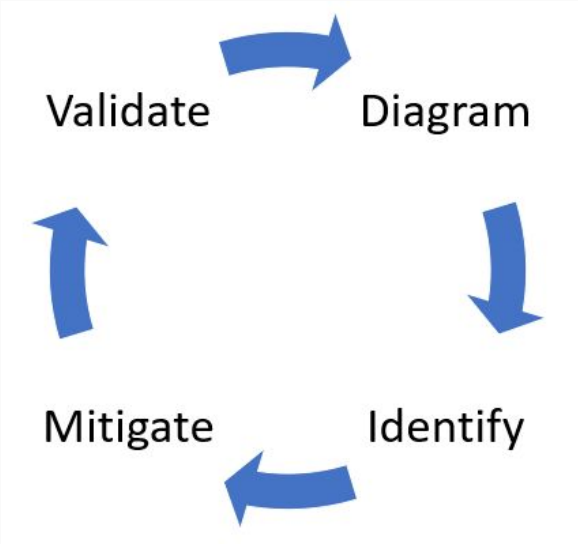
https://techbeacon.com/security/how-build-devsecops-grow-culture-ground

DevSecOps Model



Threat Model

- Threat modeling is a structured approach of **identifying and prioritizing potential threats** to a system, and determining the value that **potential mitigations** would have in reducing or neutralizing those threats. - OWASP
- Microsoft Threat Modeling Tool



Secure Code Standards

- CMU SEI - Top 10 CERT Coding Standards :
  https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices
- SEI CERT Oracle Coding Standard for Java :
  https://wiki.sei.cmu.edu/confluence/display/java/SEI+CERT+Oracle+Coding+Standard+for+Java
- Secure Coding Guidelines for Java SE:
  https://www.oracle.com/java/technologies/javase/seccodeguide.html

Static Code Analysis

- Performed without running the application
- SAST (Static Application Security Testing) : identify weakness that leads to security vulnerability.
  - Static Source Code Analysis : IntelliJ's native Code Analyzer, CodeMR, SpotBugs, PMDPlugin
  - Linters : CheckStyles, SonarLint, KLint, JSLin

- SCA (Software Composition Analysis)
  - looks for the open source components against known vulnerability.
    - BOM (Bill of Material) of the codebase is compared against Database of vulnerabilities eg. National Vulnerability Database (NVD).
  - also evaluates security, license compliance, and code quality

Dynamic Code Analysis

- Performed on running the application
- Dynamic Application Security Testing (DAST)
- Mostly used in the context of WebApps
- Web Application Vulnerability Scanners
- Fuzzers
- Attack proxies
  - Zed-attack proxy (ZAP),
  - Burp Suite (PortSwigger)