

# SDLC Trends

MIS5903

<https://community.mis.temple.edu/mis5903sec711summer2021/>

1

---

---

---

---

---

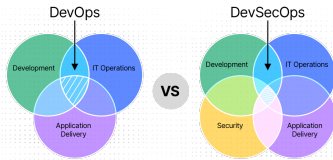
---

---

---

## Why DevSecOps?

- Reduce vulnerabilities by introducing security earlier into the project.
- “Security as Code”
  - Developers consider security issues
  - Security teams automate tasks
- Communication
- Collaboration <https://www.imperva.com/learn/application-security/devsecops-devops-security/>



2

---

---

---

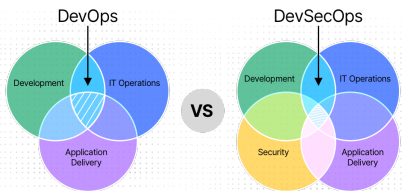
---

---

---

---

---



- Developers and IT Operations work together
- Add Security Considerations to the development pipeline

3

---

---

---

---

---

---

---

---

### DevOps Elements - Microservices

- Software is a set of independent services
- Can run in their own VM or container
- Easier to identify and support

---

---

---

---

---

---

---

4

### DevOps Elements – Infrastructure as Code (IaC)

- code manages and automates computing resources such as hosts, virtual machines and containers.
- Reduces the need for IT assistance and/or supervision.
- Spin up environments on demand

---

---

---

---

---

---

---

5

### DevOps Elements – Policy as Code (PaC)

- Manage organizational policies
  - Technologies
  - Security standards
  - IT Practices
- Policies provided in code format
- Easily enforce across development stages

---

---

---

---

---

---

---

6

### DevSecOps – Shifting Security Left

- “shifting left” – completing a task earlier
- Security embedded in the codebase earlier
- Functional testing is not the minimum requirement
- Tested code must be free of security flaws and/or vulnerabilities

---

---

---

---

---

---

---

7

### DevSecOps – Continuous Feedback Loop

- Encourages developers to improve development and maintenance practices
- Automated processes
  - Real-time alerts
- Teams can collaborate to fix immediately.

---

---

---

---

---

---

---

8

### DevSecOps - Automation

- Automated Code Analysis
- Compliance Monitoring
- Threat Investigation

---

---

---

---

---

---

---

9

### CI/CD

- Continuous Integration
  - Developers merge code changes back to a shared branch or trunk
- Continuous Delivery
  - Codebase ready to be deployed
- Continuous Deployment
  - Deploy smaller changes
  - Deploy quicker, especially for cloud-based applications

---

---

---

---

---

---

---

---

10

### Elements of a CI/CD pipeline

- Build
- Test
- Release (repository. E.g. Github)
- Deploy (production)
- Validation and Compliance




---

---

---

---

---

---

---

---

11

### CI/CD Tools

- Jenkins
- Tekton Pipelines
- Open Source:
  - Spinnaker
  - GoCD
  - Concourse
  - Screwdriver
- Managed CI/CD:
  - GitLab
  - CircleCI
  - Travis CI
  - Atlassian Bamboo
- Automation:
  - Ansible
  - Chef
  - Puppet

---

---

---

---

---

---

---

---

12

## Containers

- Container Runtimes
  - Docker
  - Rkt
  - Cri-o
- Container Orchestration
  - Kubernetes

---

---

---

---

---

---

---

13

## Questions?

---

---

---

---

---

---

---

14