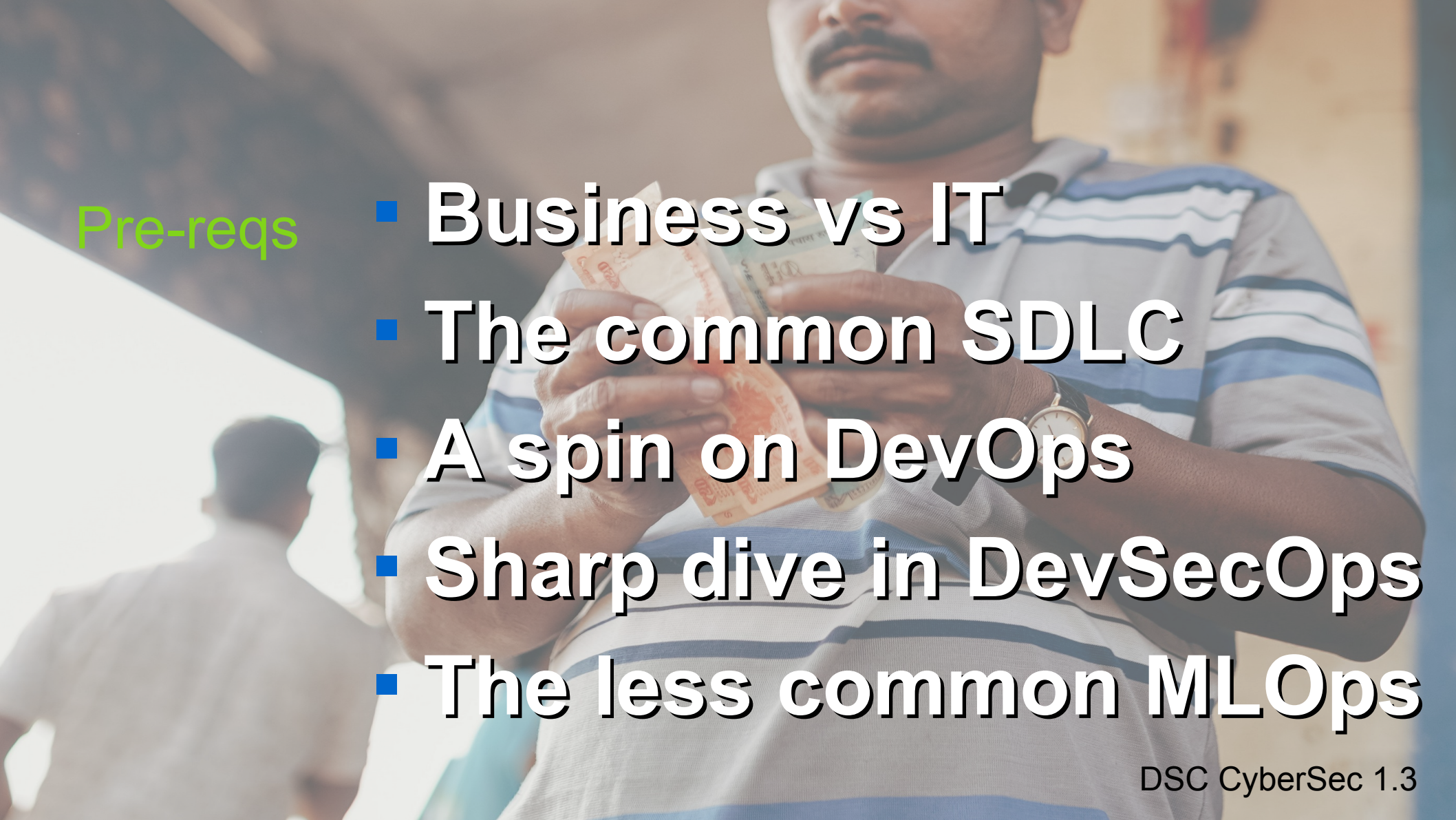

CODE SECURITY
with
DSC Cyber Division
CYBER SECURITY - CYBER SECURITY - CYBER SECURITY - CYBER SECURITY -
DSC CyberSec 1.3

“There is a Cyberattack Every 39 Seconds.”

”43% of cyber attacks are targeted at small businesses.”

” 60% of small businesses that suffer a cyber attack go out of business within six months.”



Pre-reqs

- Business vs IT
- The common SDLC
- A spin on DevOps
- Sharp dive in DevSecOps
- The less common MLOps

No chaos today!

Disclaimer

Echo ”uname -a”

```
.global _start
_start:
        mov r7,     #4
        mov r0,     #1
        ldr r1,     =Captain
        mov r2,     #9
        svc 0

        mov r7,     #1
        mov r0,     #0
        svc 0
.data
Captain:  .ascii  "D_captain!\n"
```

D_captain

- Cybersecurity enthusiast
  (Web apps, Networks, Digital Forencics)
- Python Developer
- GNU fan
- Technical writer
- CTF player @Fr334aks-Mini
- Builder @TheShield
- And many other things:
  Shotokan, Bikes, Philosophy, Astronomy $ Planetary science

https://dennismasila.github.io

D_captainkenya

DSC CyberSec 1.3

Business and IT Processes

What Is Business Process?

- A business process is a series of interlinked steps which are assigned to every stakeholder for a specific work to deliver a product or service to the customer.

Business and IT Processes

Examples of Business Process

- Product Development
- Manufacturing
- Delivery
- Sales
- Marketing
- Customer Service
- Accounting
- Maintenance
- Management
- Finance
- Onboarding

Business and IT Processes

Types of Business Process

- Primary Processes: Fundamental processes of a business through which a company delivers the end product to the customer.
- Support Processes: Don’t add value to the final product directly but they make an environment for primary processes to operate efficiently and effectively.
- Management processes: Management processes govern operations, corporate governance and strategic management. These processes set goals and standards which lead to the efficient and effective working of primary and support processes. Besides planning, these processes also involve monitoring and control of other business processes.

Business and IT Processes

Importance Of Business Process

- Reduced expenditure and risk
- Reduce human error
- Improving efficiency
- More customer focused
- Bridging communication gaps
- Better time management
- Adaption of new technology

Business and IT Processes

IT Process(Operations)

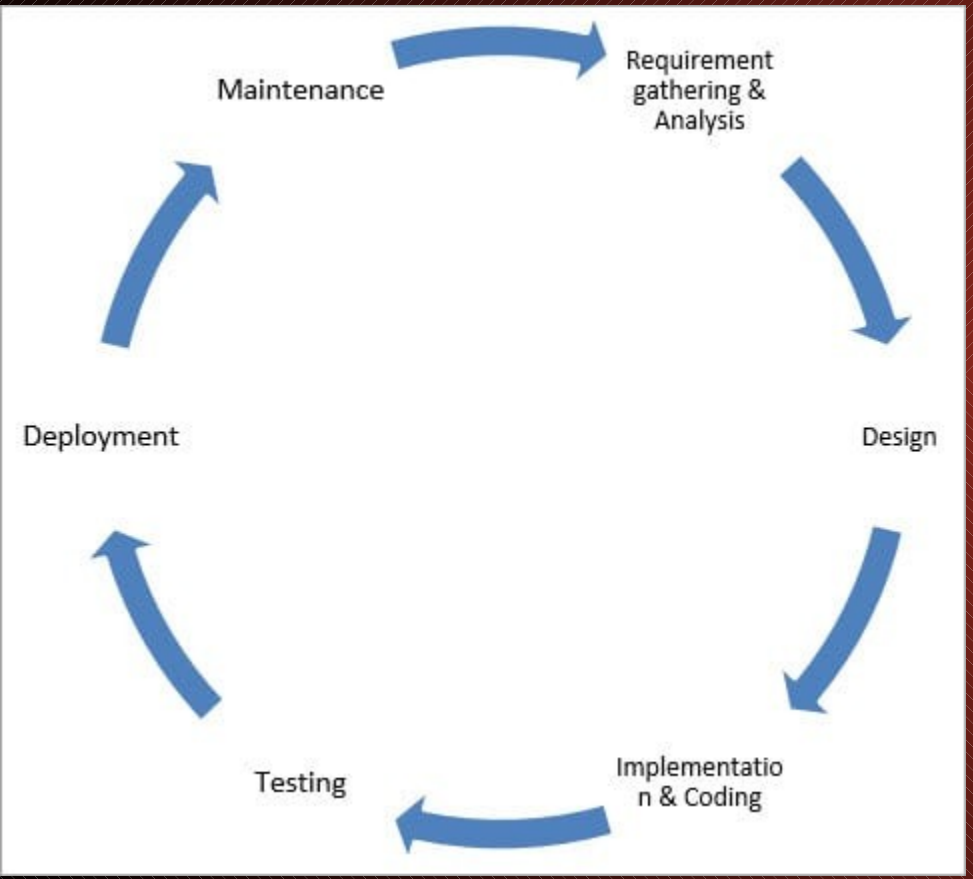
- The process of implementing, managing, delivering and supporting IT services to meet the business needs of internal and external users.
- Roles::
- IT support,
- Incident response and Security enforcement,
- Ensuring application performance,
- Optimizing IT infrastructure,
- Managing resources.

Business and IT Processes

IT Operations Examples

- IT asset management (ITAM)
- Backup and recovery
- Security analysis
- Data warehouse analysis
- Business continuity planning
- Software development
- Testing
- Configuration management
- Vendor management

SDLC – The Common Framework!



The what?

- A process that defines the various stages involved in the development of software for delivering a high-quality product.
- SDLC stages cover the complete life cycle of a software i.e. from inception to retirement of the product.
- Optimized by DevOps

A Spin on DevOps

- A compound of development (Dev) and operations (Ops), DevOps is the union of people, process, and technology to continually provide value to customers.

- DevOps enables formerly siloed roles—development, IT operations, quality engineering, and security—to coordinate and collaborate to produce better, more reliable products.

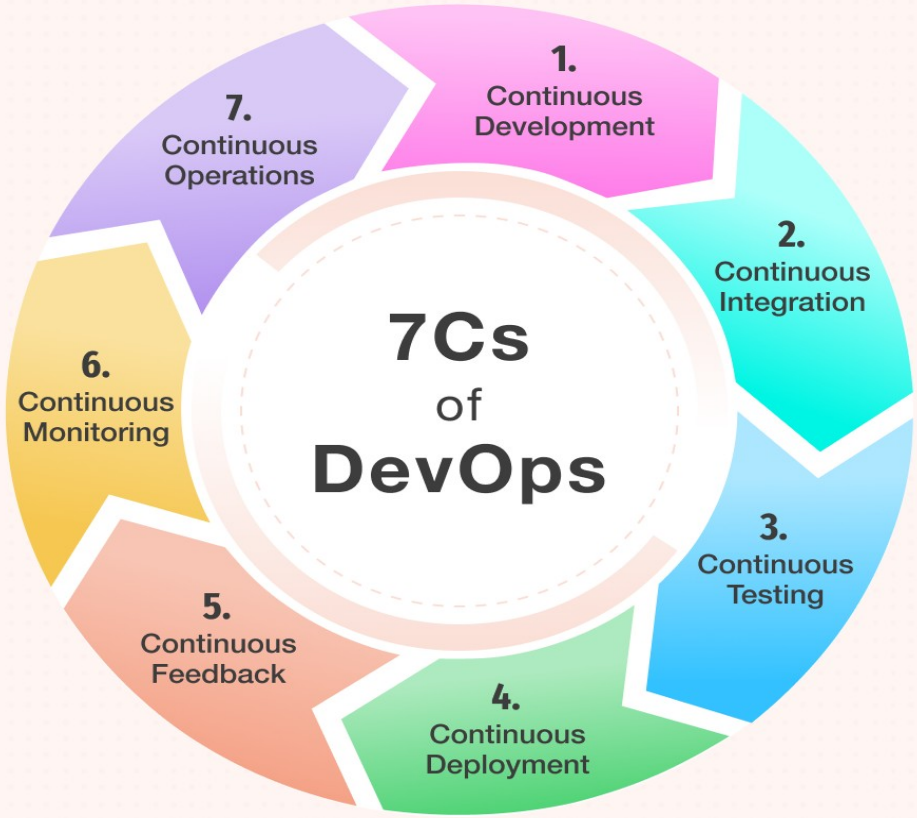
SDLC vs DevOps – Unrelated, right?

SDLC is incompatible with a DevOps approach, which seeks to reduce the size of batches and increase collaboration between different disciplines.

- A Using an SDLC, you'd arrange 20 people into 5 specialist teams to work on phases such as analysis, design, development, testing, and operations. These horizontal teams would perform their specialist task with work passed from team to team, like the baton in a relay race.

- In DevOps, you'd arrange people into 4 cross-functional teams who could deliver software without hand-offs. Your vertical teams could each deliver and run an isolated component, like the line of players moving the ball toward the scoring line in a game of rugby.

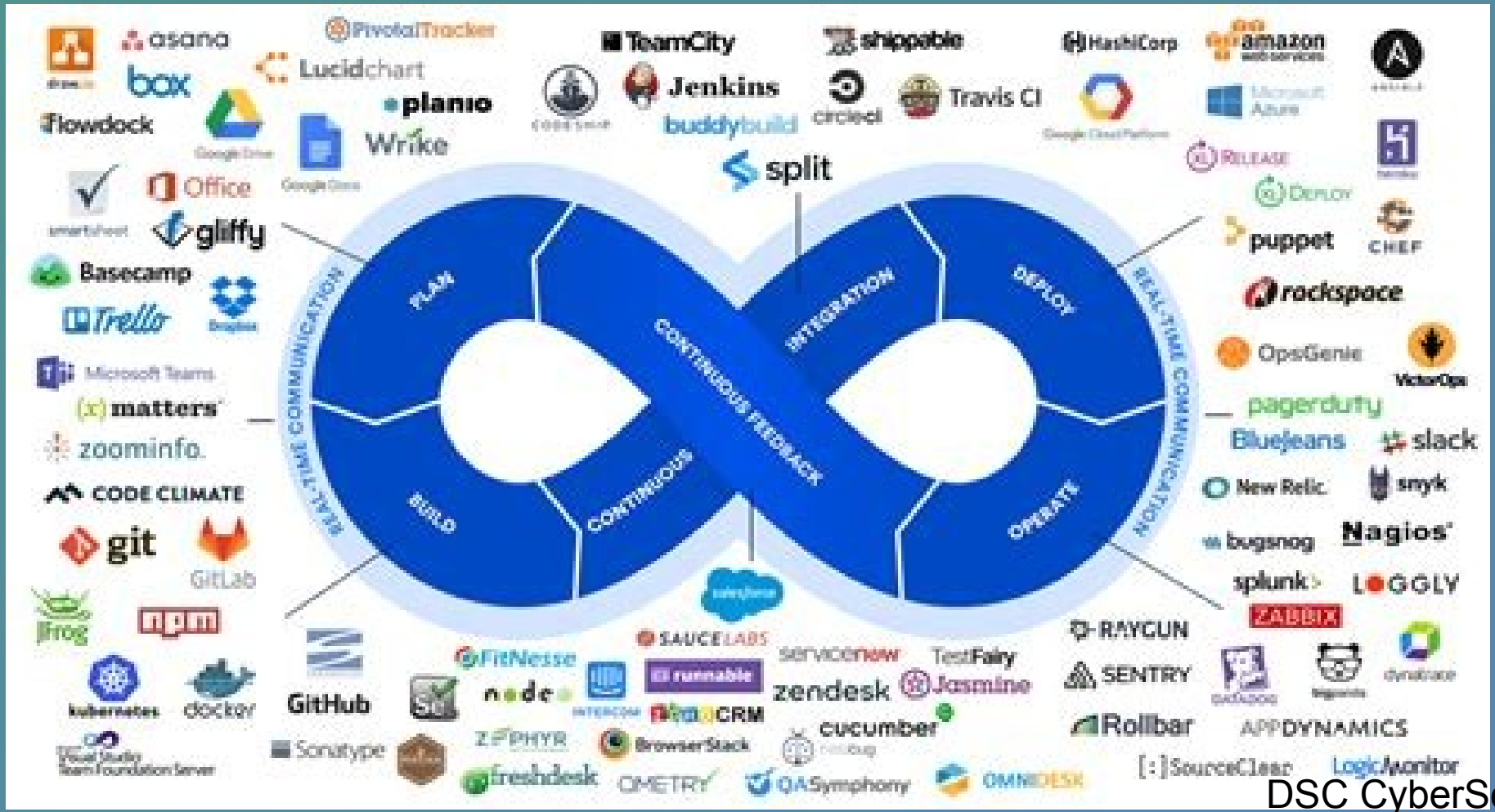
DevOps Lifecycle(pipelines)



Benefits of DevOps

- Speed
- Rapid Delivery
- Reliability
- Scale
- Improved Collaboration
- Security

DevOps Tools



	Tools	
DevOps Phases	**Continuous Planning & Development**	• GitLab • GIT • TFS • SVN • Mercurial • Jira • BitBucket • Trello • Maven • Gradle • Confluence • Subversion • Scrum • Lean • Kanban
	Continuous Integration	• Jenkin • Bamboo • GitLab CI • TeamCity • Travis and CircleCI • Buddy
	Continuous Testing	• JUnit • Selenium • JMeter • Cucumber • TestSigma • Microfocus UFT • TestNG • Tricentis Tosca • Jasmine
	Continuous Deployment	• Ansible • Chef • Docker • IBM Urban Code • Kubernetes • Puppet • Go • Vagrant • Spinnaker • ArgoCD
	Continuous Monitoring	• Nagois • Grafa na • Kibana • Prometheus • Logstash • AppDynamics • ELK Stack • New Relic • Splunk • Sensu • PagerDuty
	Customer Feedback	• Webalizer • W3Perl • ServiceNow • Slack • Flowdock • Open Web Analytics • Pendo • Qentelli's TED
	Continuous Operations	• Kubernetes • Docker Swarm

Tools Across DevOps Phases

Sharp dive into DevSecOps

A short for ~~development, security, and operations~~—automates the integration of **security at every phase** of the software development lifecycle, from initial design through integration, testing, deployment, and software delivery.

It integrates **application and infrastructure security** seamlessly into Agile and DevOps processes and tools. It addresses security issues as they emerge, when they're easier, faster, and less expensive to fix (and before they are put into production).

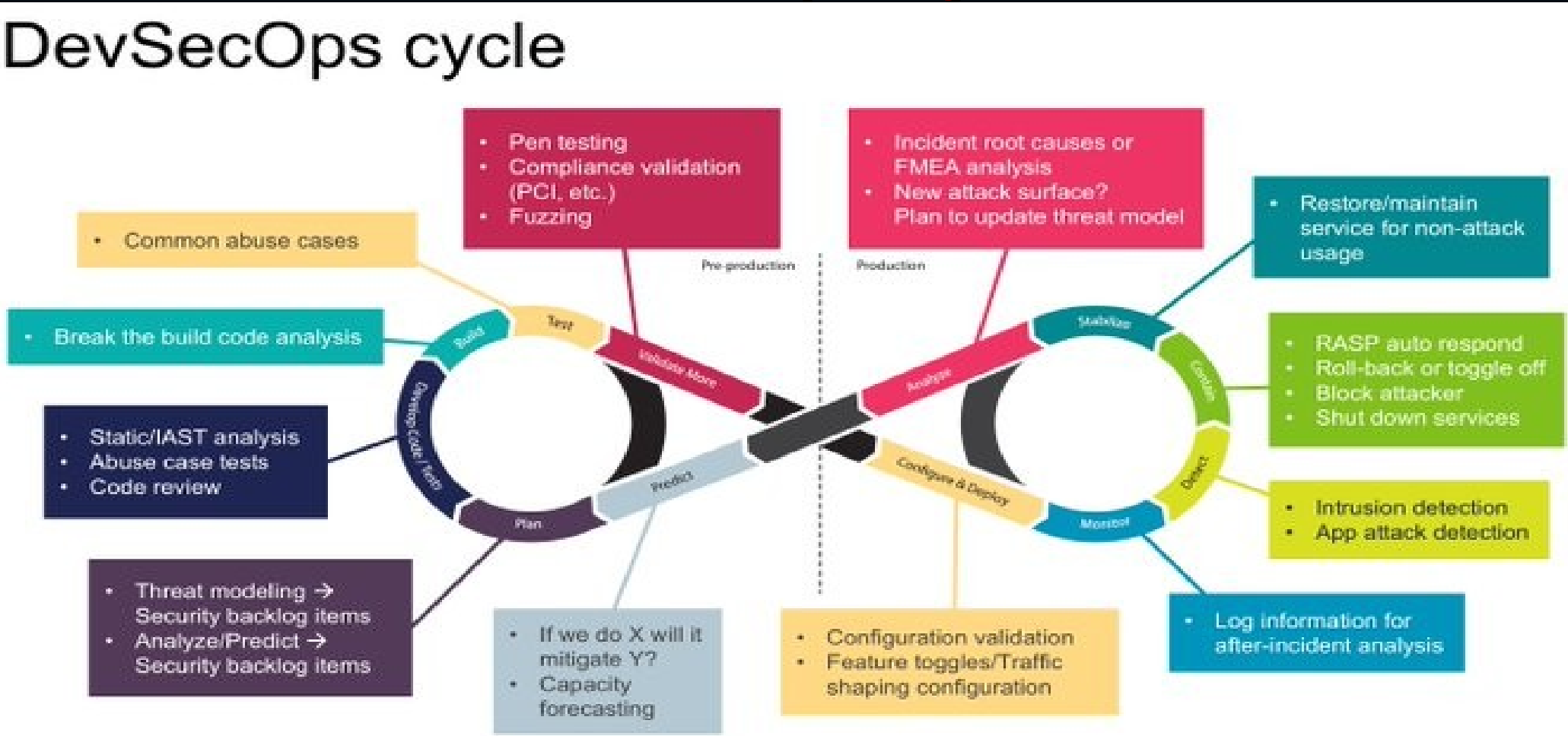
Application Security

Application security is the use of software, hardware, and procedural methods to protect applications from external threats.

Modern approaches include shifting left, or finding and fixing vulnerabilities earlier in the development process, as well as shifting right to protect applications and their infrastructure-as-code in production.

DevSecOps makes application and infrastructure security a shared responsibility of development, security, and IT operations teams, rather than the sole responsibility of a security silo..

DevSecOps Cycle



Key take-away

Shifting left

'Shift left' is a DevSecOps mantra: It encourages software engineers to move security from the right (end) to the left (beginning) of the DevOps (delivery) process. .

DevSecOps brings in cybersecurity architects and engineers as part of the development team. This ensures every component, and configuration item is patched, configured securely, and documented.

Not only is the development team building the product efficiently, but they are also implementing security as they build it.

Benefits of DevSecOps

Rapid, cost-effective software delivery

Improved, proactive security

Accelerated security vulnerability patching

Automation compatible with modern development

A repeatable and adaptive process.

The To do's

Environment and data security

* Standardize and automate the environment(Zero trust)

* Centralize user identity and access control capabilities

* Isolate containers running microservices from each other and the network

* Encrypt data between apps and services

* Introduce secure API gateways

The To do's

CI/CD pipeline security

* Integrate security scanners for containers

* Automate security testing in the CI process

* Add automated tests for security capabilities into the acceptance test process

* Automate security updates, such as patches for known vulnerabilities

* Automate system and service configuration management capabilities

Hands on – code surity

Reqs:

* Github account
* VS Code/ Codium
* Email
* A brain, or the mind!

Hands on – code security

Reqs:

* Basic Security
* 2FA
* ENV Variables
* Github security(each repo)Dependabot
* Gitguardian
* VS Code – Snyk security

Hands on – code security

* **Basic Security**
  - Require passwords in devices at workplace.
  - Enforce lockouts after inactivity.

Hands on – code security

* 2FA – Multi-factor authentication
  - Developer accounts
    - Github:
      settings-access-passwd/auth-2fa(auth app)

Hands on – code security

* **ENV Variables and secrets**
  - No hard-coded env variables
    API_KEY = xxxxxxxxxxxxx
    SECRET = xxxxxxxxxxxxx
  - .env files
  - On Github:
    repo-settings-security-secrets and variables

Hands on – code security

* Github repo security Dependabot:

- scans code dependencies for vulnerabilities
- creates alerts(github notifications)

all - settings-account-security-code security and analysis
repo - repo-settings-security-code security and analysis

Hands on – code security

* Gitguardian

- Detect hard-coded secrets in commits and repositories
- Install as app
  repo – repo-settings-intergrations-github apps
- sends mail
- awesome dashbord for resolves

Hands on – code security

* security logs

- Monitor github activity
  - Logins
  - Repo activity

settings-archives-security log

Hands on – code security

* Snyk VS Code extension

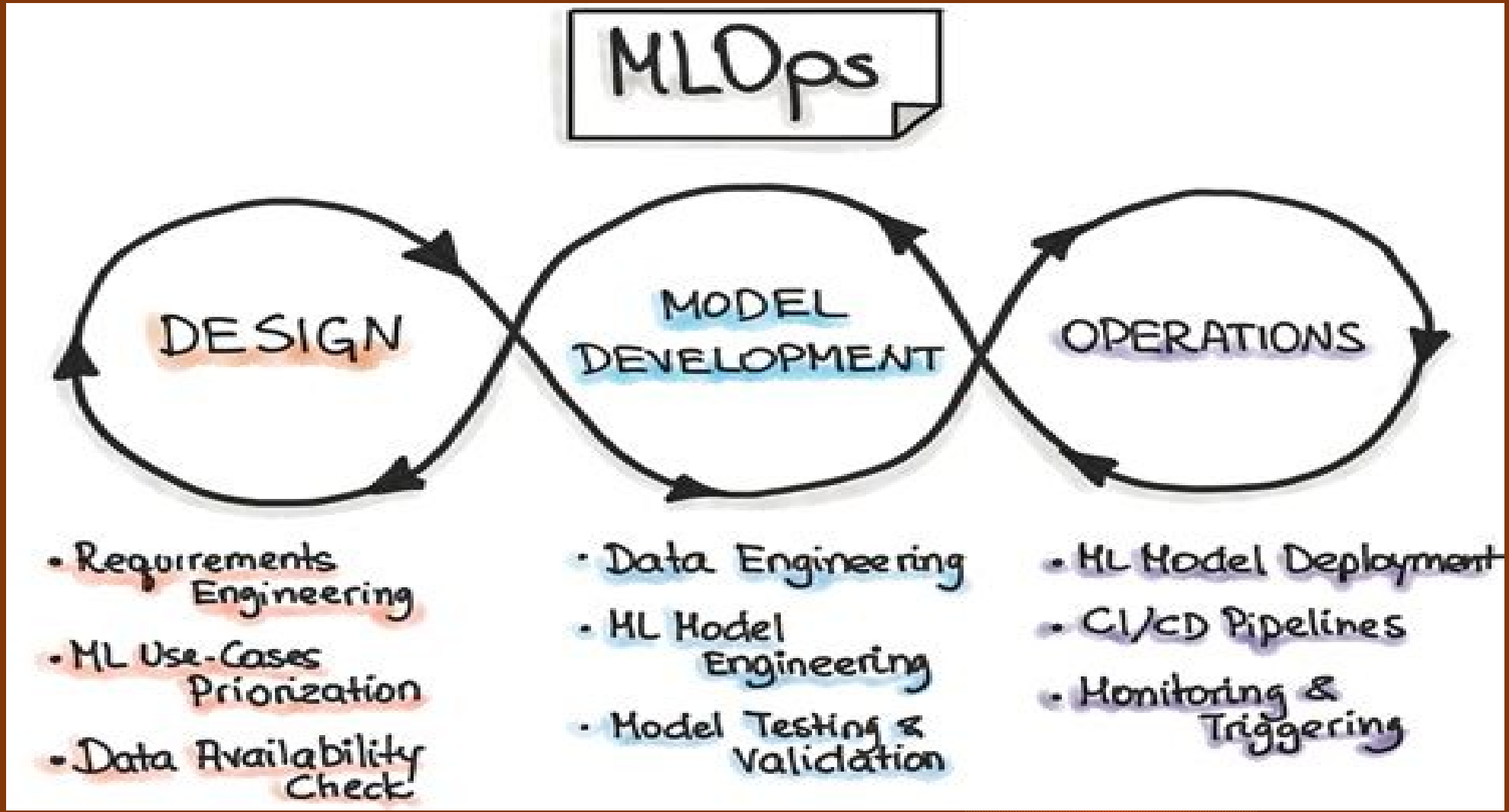
- Scans for security vulnerabilities and license issues
- Configuration issues in your IaC templates:
  Terraform,
  Kubernetes,
  Cloud Formation,
  Azure Resource Manager.

MLOps

A combination of DevOps, data engineering, and ML techniques.

Deploy, and maintain machine learning (ML) systems reliably and efficiently using the process of MLOps

This involves training, deploying, and maintaining machine learning models to ensure efficiency. Security is an essential component of all MLOps lifecycle stages.


MLOps
DESIGN
MODEL DEVELOPMENT
OPERATIONS
• Requirements Engineering
• ML Use-Cases Prioritization
• Data Availability Check
• Data Engineering
• ML Model Engineering
• Model Testing & Validation
• ML Model Deployment
• CI/CD Pipelines
• Monitoring & Triggering

MLOps - Security

Protecting data storage:

Zero trust policy(Risk based auth)
The principle of least privilege (PloP)
Monitoring and logging data storage access

Securing ML models:

Data poisoning - Attackers manipulate training data to ensure the resultant ML model is vulnerable to attacks.
Validity checking - before training an ML model.

MLOps

Compliance policies:

ML models using sensitive or private data: PII, Patient data.
Consent during data collection.
Compliance with authorities governing the use of sensitive patient data.

Observability and logging of ML tasks:

Observability - seeks to understand the ML system in its healthy and unhealthy states.
Prevents failures by providing alerts before an incident occurs and recommending solutions for those failures.
Logging – Track performance data and metrics of the ML tasks

Q and A

Reference:

- Trendmicro.com
- Amazon.com
- Simform.com
- Ibm.com
- Redhat.com

DSC Cyber Division