Leveraging DevOps and DevSecOps to Accelerate AI Development and Deployment

Hasan Yasar

Technical Director, Adjunct Faculty Member

Software Engineering Institute | Carnegie Mellon University

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0810

Outline

- Modern SW Development: DevOps
  - Fundamentals,
  - DoD Specific -ATO
- AI for DevOps
- DevOps for AI

Modern SW Development: DevOps

DevOps is a set of principles and practices emphasizing collaboration and communication between software development teams and IT operations staff along with acquirers, suppliers, and other stakeholders in the lifecycle of a software system

DevSecOps is a model on integrating the software development and operational process considering security activities: requirements, design, coding, testing , delivery , deployment and incident response.

Mature DevOps practices are constantly testing, deploying and validating that software meets every requirement and allows for fast recovery in the event of a problem. As a result we can easily say,

“DevSecOps is DevOps done right”

Who are Dev?

- Follow Agile methodologies
  - Using Scrum, Kanban and modern development approaches
  - Self directing, self managed, self organized
- Using any new technology
  - Each Dev has own development strategy
  - OpenSource,
- Allowed to have
  - Close relationships with the business
  - Software driven economy

Want to deliver software faster with new requirements…

Who are Ops?

- Operations
  - Runs the application
  - Manages the infrastructure
  - Support the applications
- Operations provides
  - Service Strategy
  - Service Design
  - Service Transition
  - Service Operations
  - Secure systems

Want to maintain stability, reliability and security…

DevOps aims to Increase…

…the pace of **innovation**

…**responsiveness** to business needs

…**collaboration**

…software **stability and quality**

… continuous feedback

DevOps has four Fundamental Principles

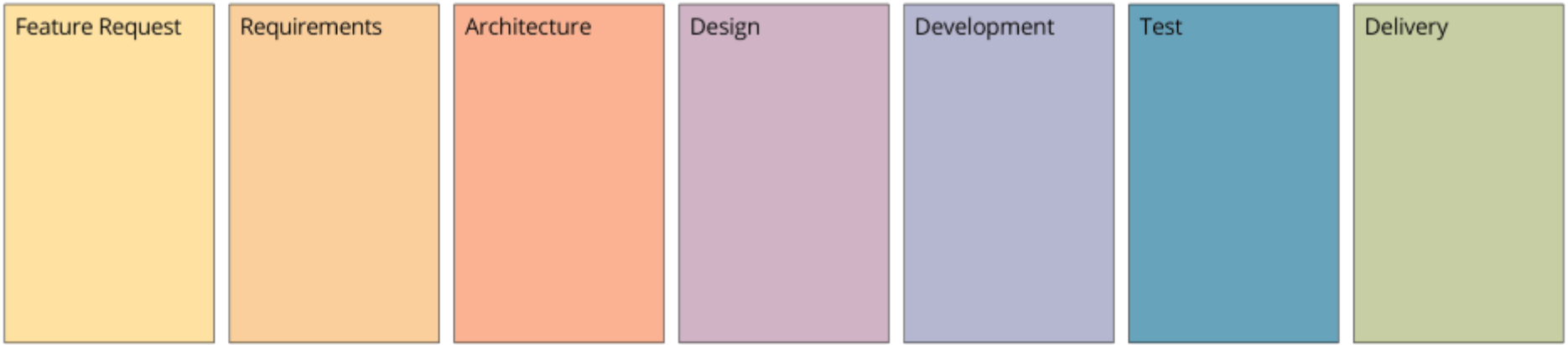
- **Collaboration:** between project team roles

- **Infrastructure as Code:** all assets are versioned, scripted, and shared where possible

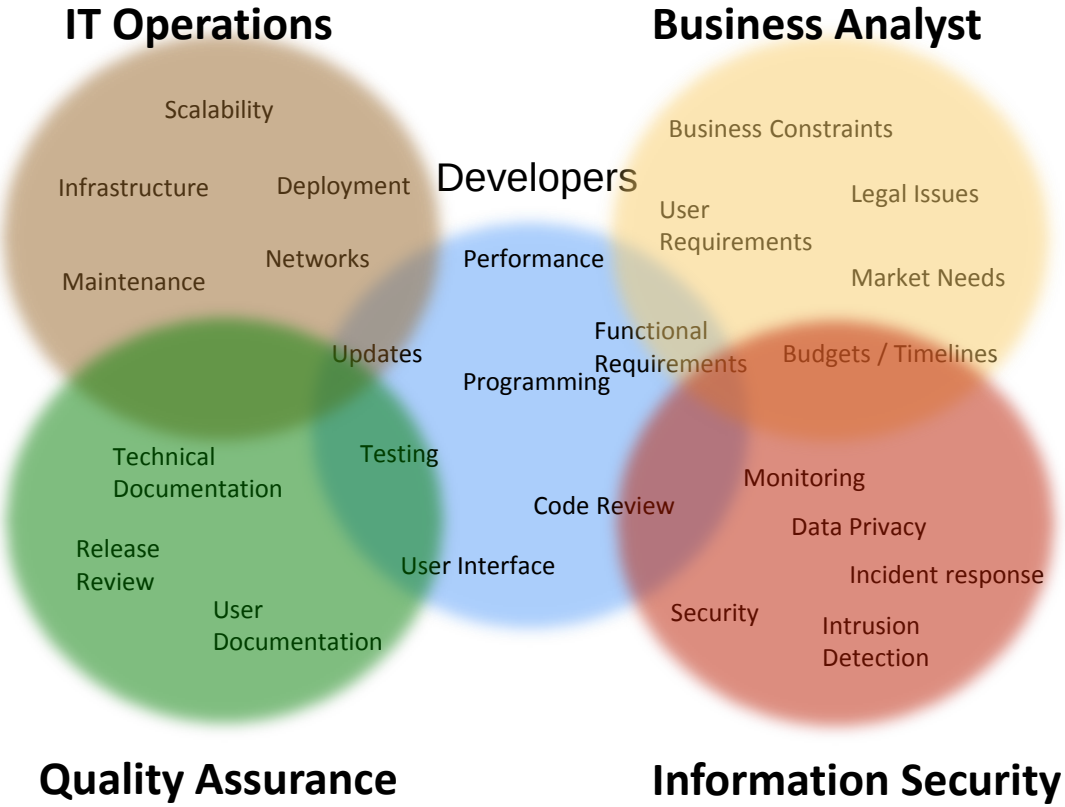
- **Automation:** deployment, testing, provisioning, any manual or human-error-prone process

- **Monitoring:** any metric in the development or operational spaces that can inform priorities, direction, and policy

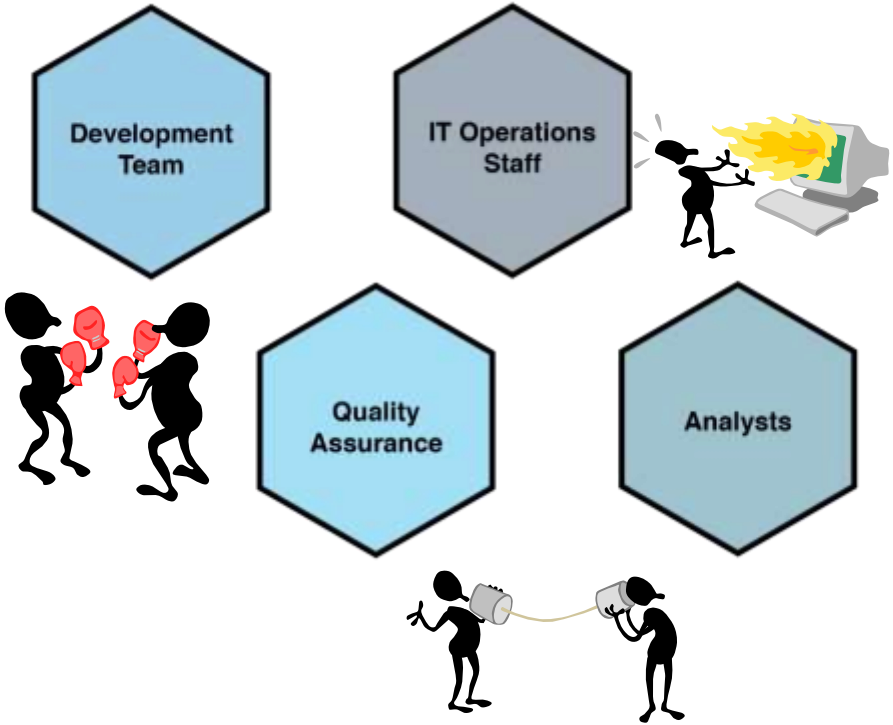
SW Development Phases



Collaboration: *Many stakeholders*

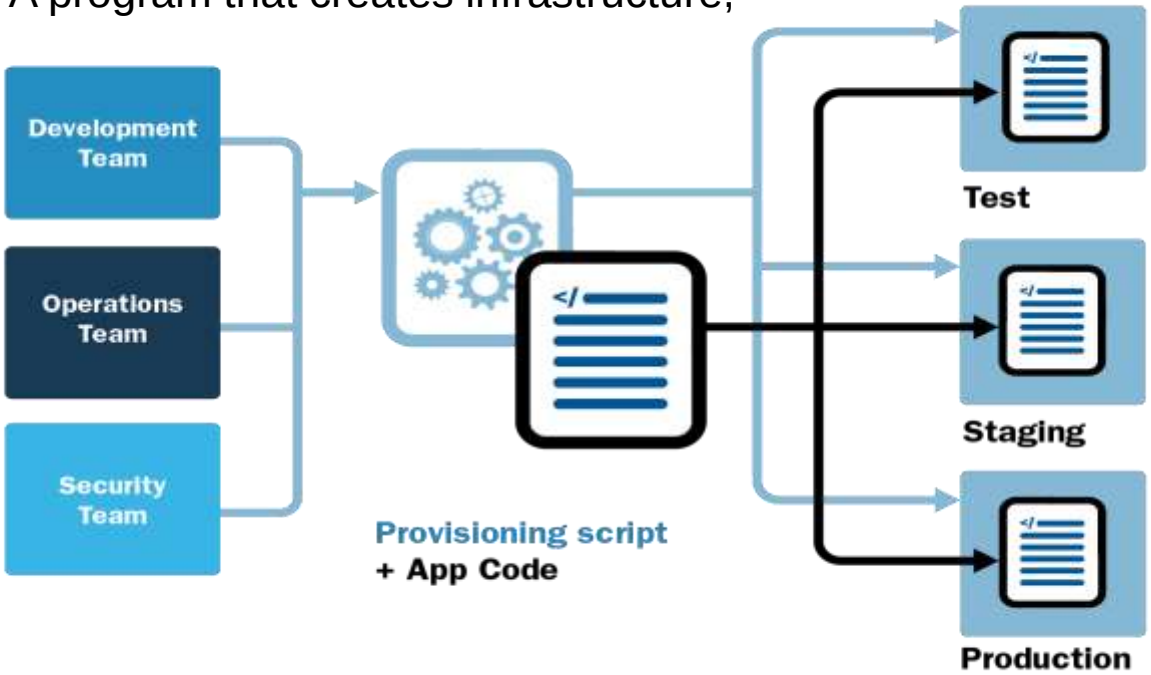


Collaboration: *Silos Inhibit Collaboration and poor communication*



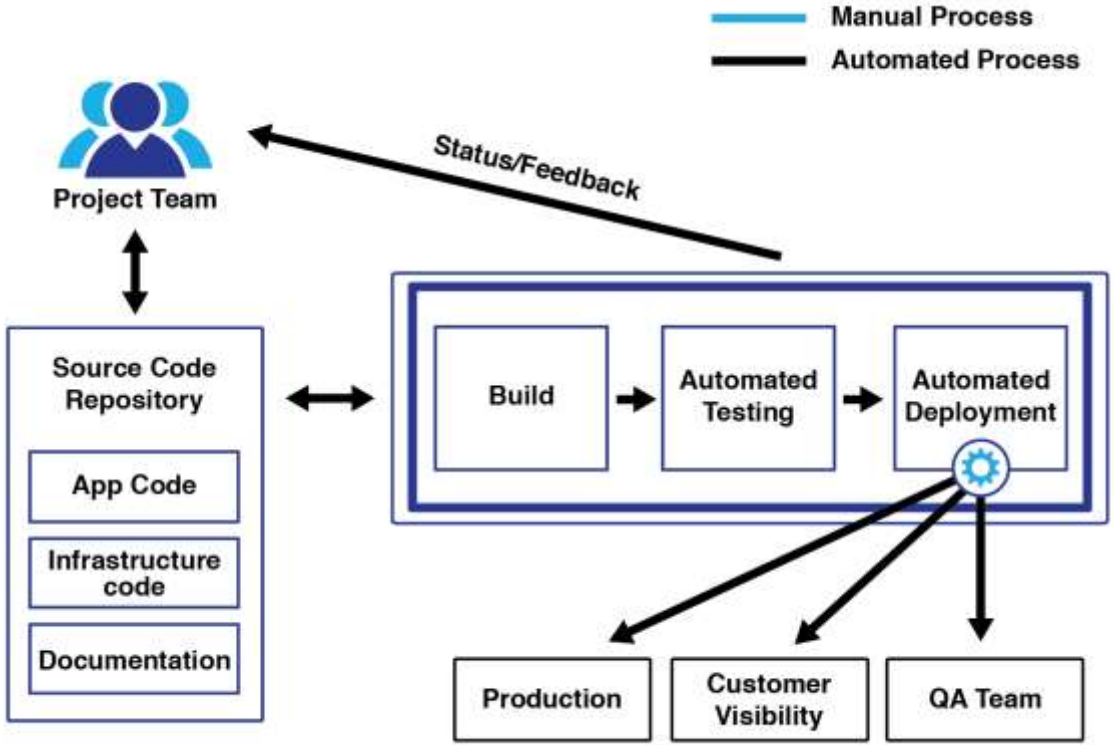
Infrastructure as Code (IaC)

A program that creates infrastructure,



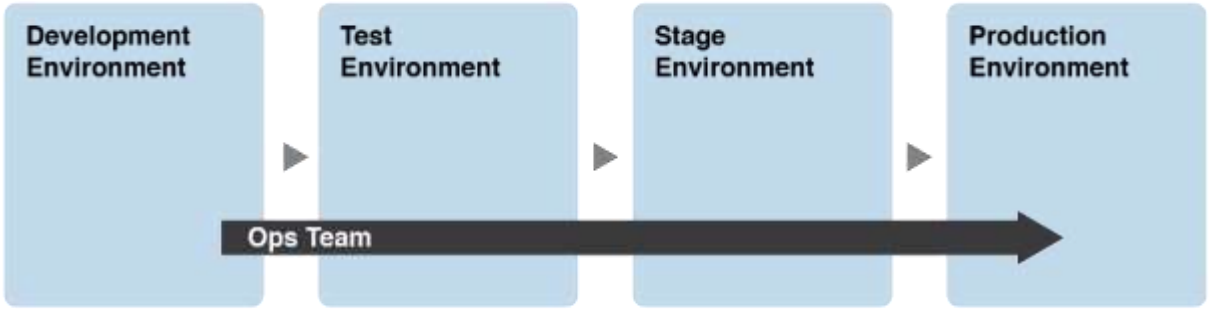
A concretely defined description of the environment is good material for conversation between team members.

Automation : *Continuous Integration (CI)*



Continuous integration is a process that continually merges a system's artifacts, including source code updates and configuration items from all stakeholders on a team, into a shared mainline to build and test the developed system.

Automation : *Continuous Delivery / Deployment (CD)*

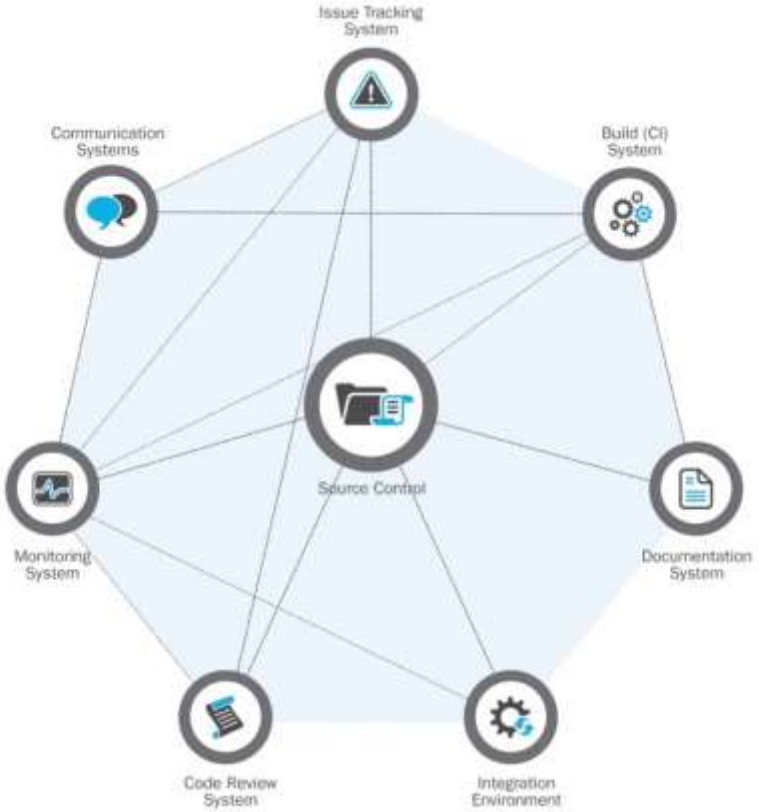


Shift Left Operational Concerns Enforced by Continuous Delivery with parity across various environment

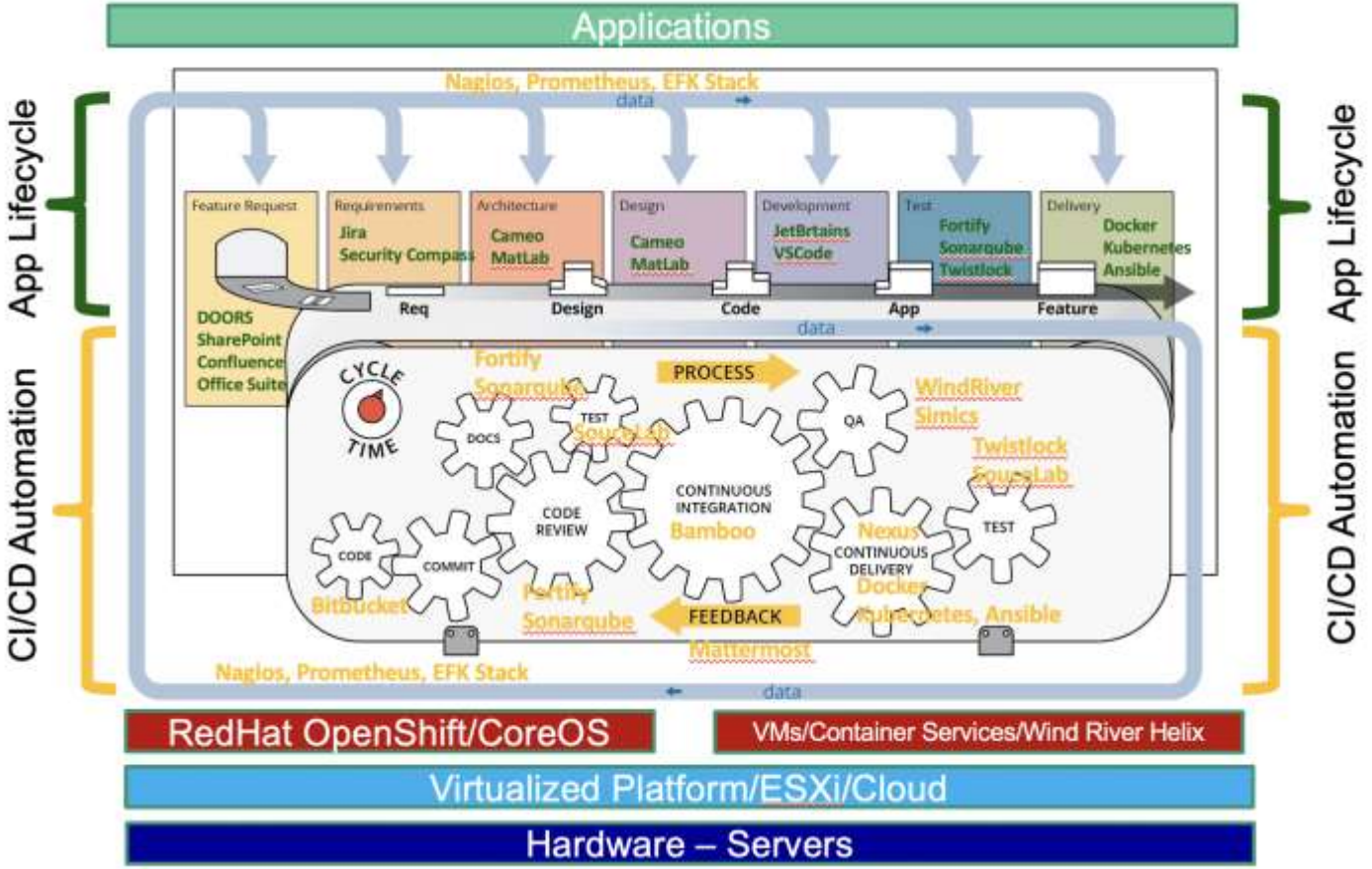
Continuous delivery is a software engineering practice that allows for frequent releases of new software to staging or various test environments through the use of automated testing.

Continuous deployment is the automated process of deploying changes to production by verifying intended features and validations to minimize risk.

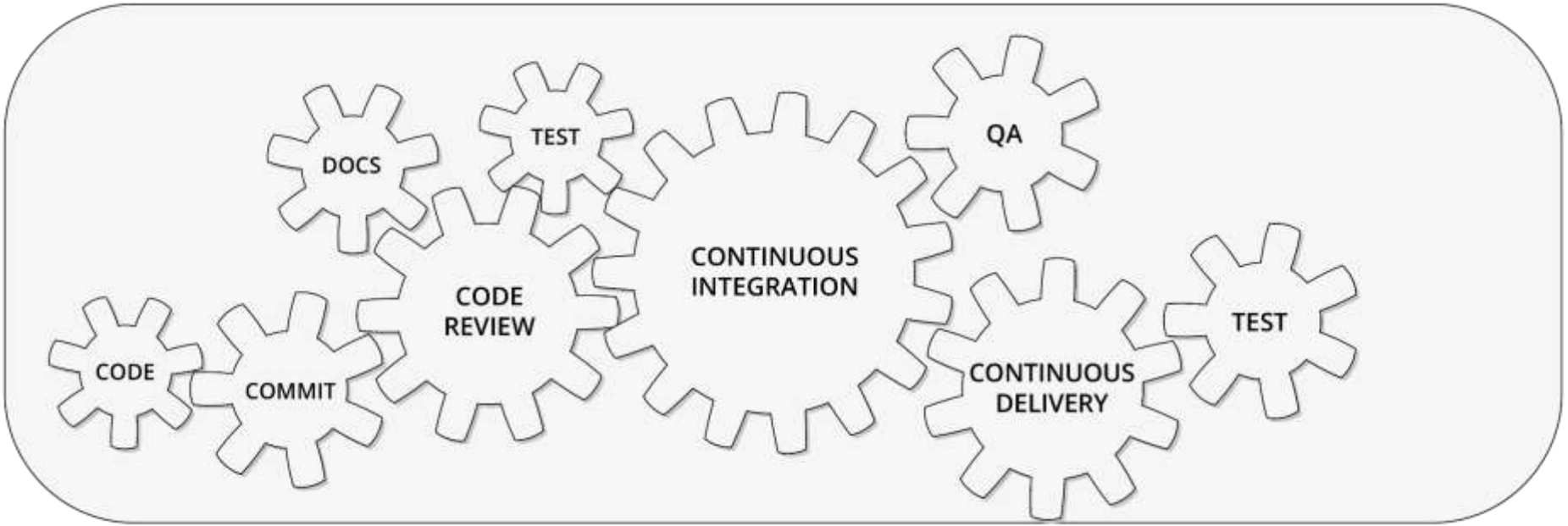
Integrated Development Pipeline - General



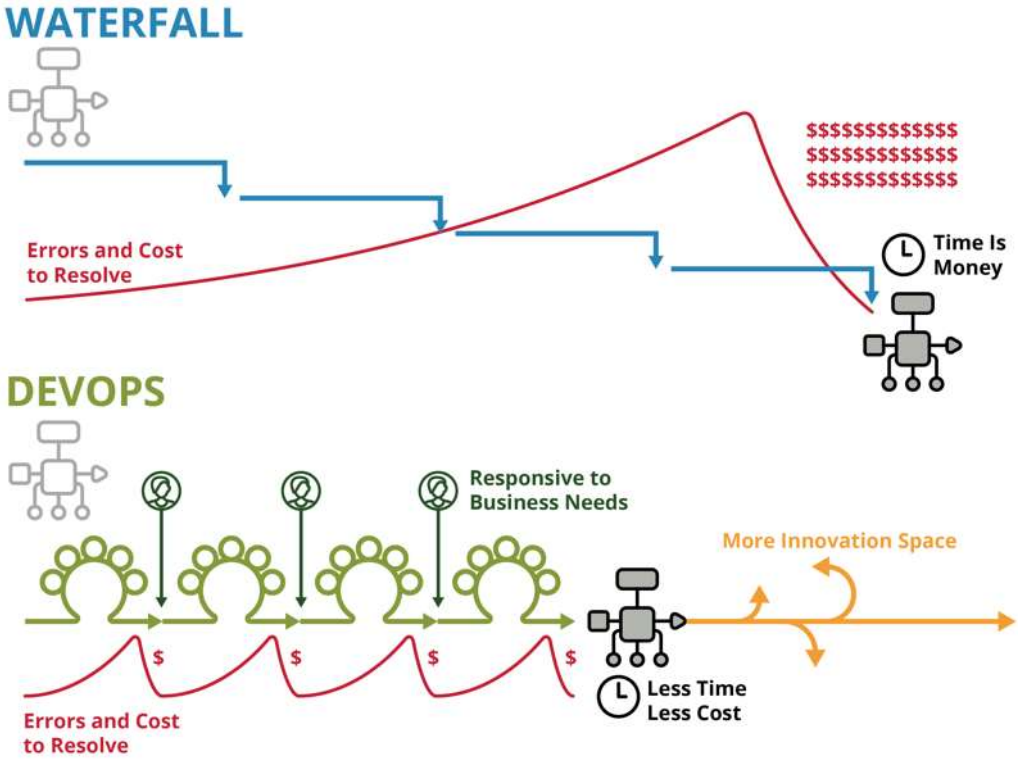
DevOps Stack: Exemplary DoD tool stack



Automation with IaC, CI, CD

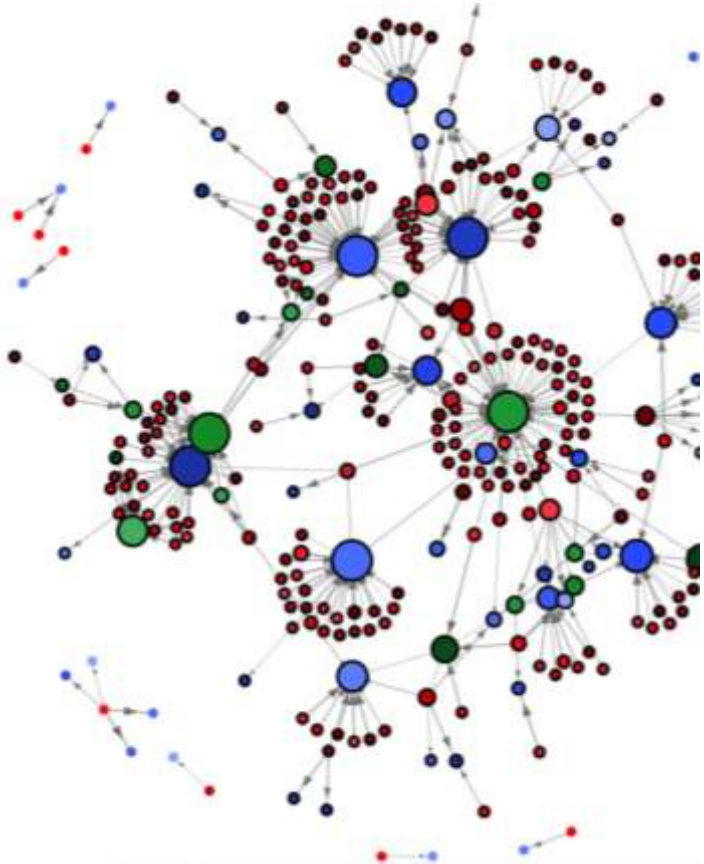


Key Benefits of DevOps



- Reduced errors during deployment
- Reduced time to deploy and resolve discovered errors
- **Repeatable** steps
- **Continuous availability** of pipeline and application
- Increased innovation time
- **Responsiveness** to business needs
- **Traceability** throughout the application lifecycle
- Increased stability and quality
- **Continuous feedback**

DoD Specific



DoD Does Not Control SW Development

- DoD does not produce most of the software it uses, but it must maintain that software
- Latent cyber vulnerabilities, those exposed during operations, and those due to underlying dependencies are putting the DoD at risk
- Finding and fixing problems late causes rework and drives up costs
- Software cost overruns are overwhelming program delivery and sustainment

Barriers to DevSecOps

- Complex systems (Safety critical, Realtime, Embedded Systems..
- Sustainment of DevOps pipeline
- Lack of iterative and incremental mindset cultural issues
- Organizational Structure
- Legacy Systems
- Lack of modular architecture, old tools/language
- Aged bureaucracy and waterfall process
- Lack of Metrics and Measurements
- Inconsistent Environments
- RMF- ATO Compliances

ATO: Authority to Operate

What is an ATO?

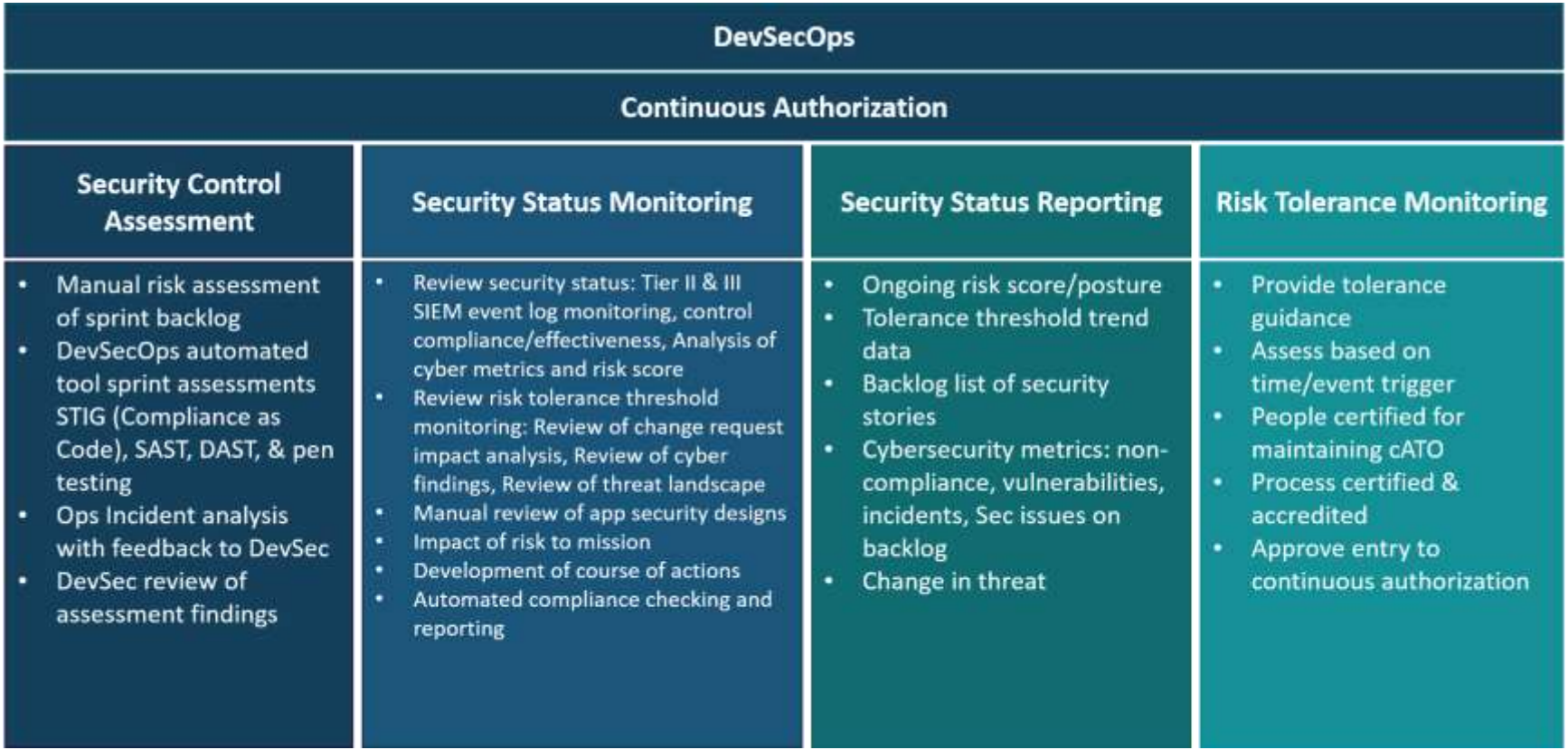
- An ATO is **Authority to Operate**
  - Authorizes the system to be placed on a production network
  - Interface with other components within the DoD
  - Authorizes access by end users to leverage these resources to execute mission
- Key staff in the ATO process
  - AO (Authorizing Official)
  - ISSO (Information System Security Officer)
  - Security assessor

- An AO makes a risk-based decision to grant an ATO for use of the system
- The decision has to be formalized in an ATO letter
  - An ATO letter must explicitly state the AO's acceptance of:
    - Use of the system at the Agency at the determined FIPS 199 impact level
    - All leveraged external services supporting the system
    - Any exceptions or exclusions of the Chief Security Officer (CSO) for use at the Agency

Continuous ATO(cATO) is the Goal

cATO authorizes the platform, process, and the team that produces the product under a continuous monitoring process that maintains the residual risk within the risk tolerance of the Authorizing Official (AO)

cATO and DevSecOps

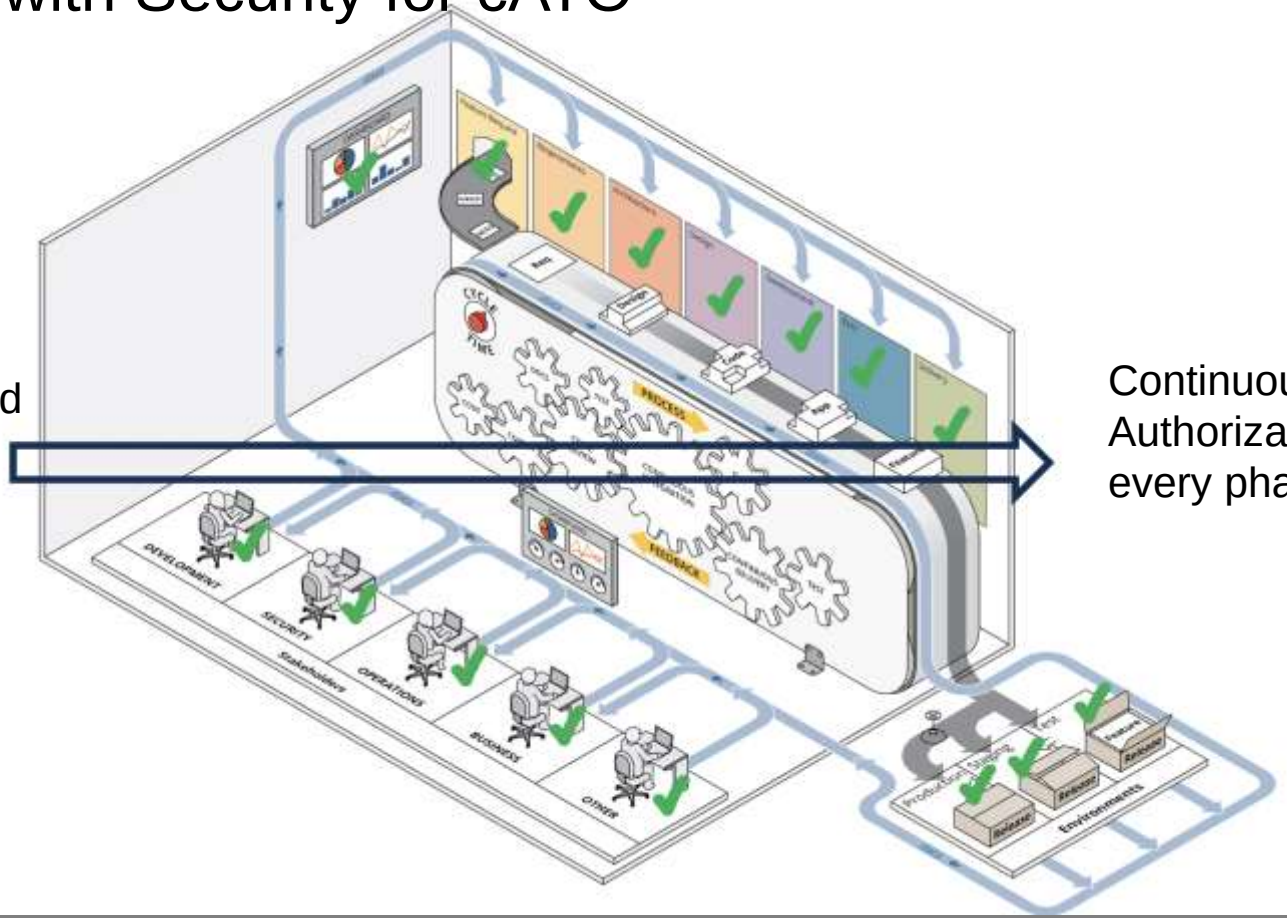


Proposed DoD Enterprise DevSecOps

- Develop and implement common DevSecOps implementation (DoD CIO DevSecOps workgroup) https://repo1.dsop.io/dsawg-devsecops
- Create and Maintain DevSecOps pipelines (and not just DevOps) to avoid each DoD service building their own stack and reinventing the wheel.
- Create hardened Container images in a dedicated artifacts repository with security built-in and compliance with FedRAMP/NIST (similar to gold images concept).
- Create a Microservice Service Architecture with Service Mesh (ISTIO).
- Standardize metrics and define acceptable thresholds for test coverage, security, documentation etc. to enable complete continuous deployment with pre-ATO embedded.
- Leverage Kubernetes for Orchestration to ensure automation, rolling-update, scale, security and visibility thanks to the sidecar security container concept.

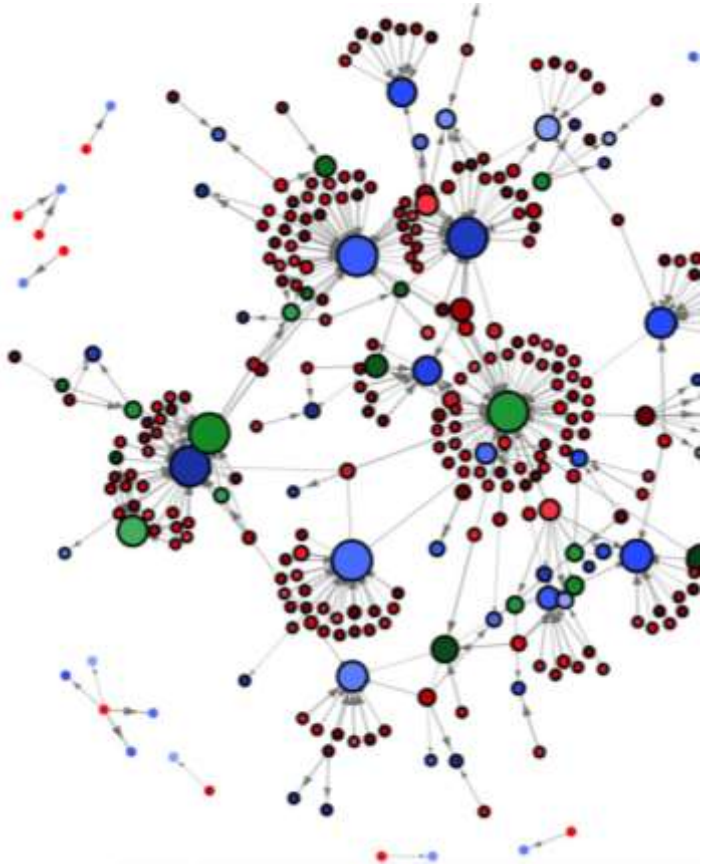
DevOps with Security for cATO

Security from inception to deployment and improvement with every delivery

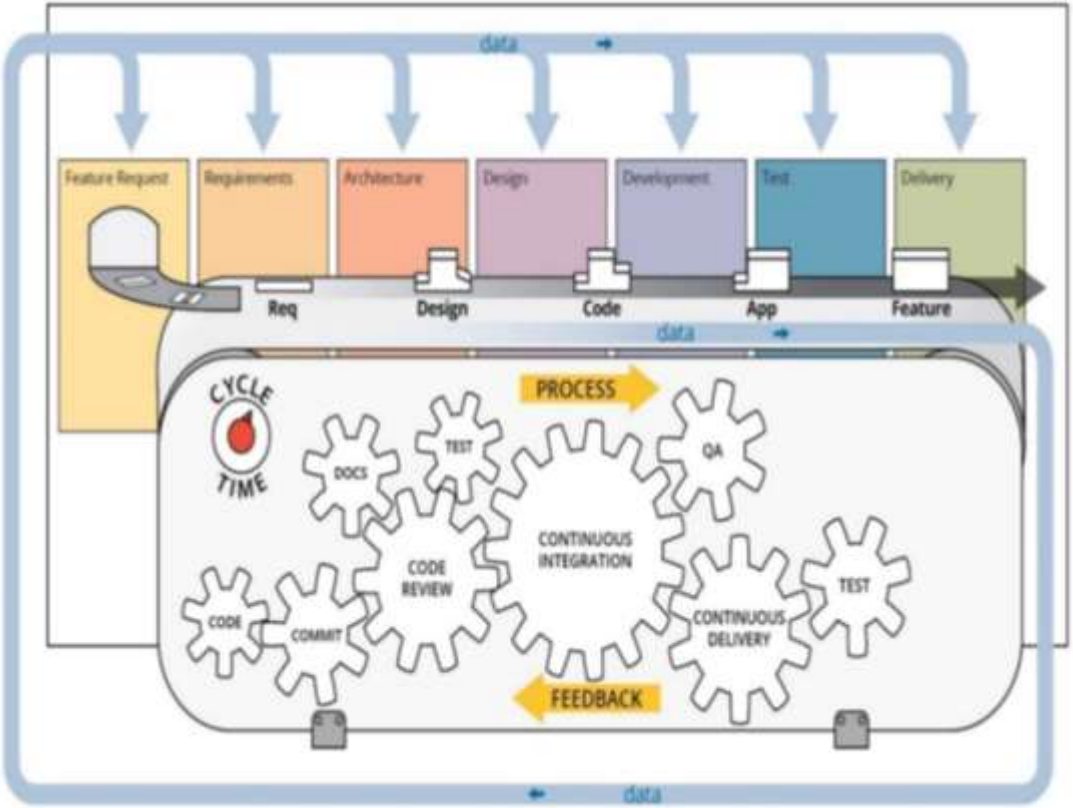


Continuous Authorization on every phases

AI for DevOps



AI For DevOps

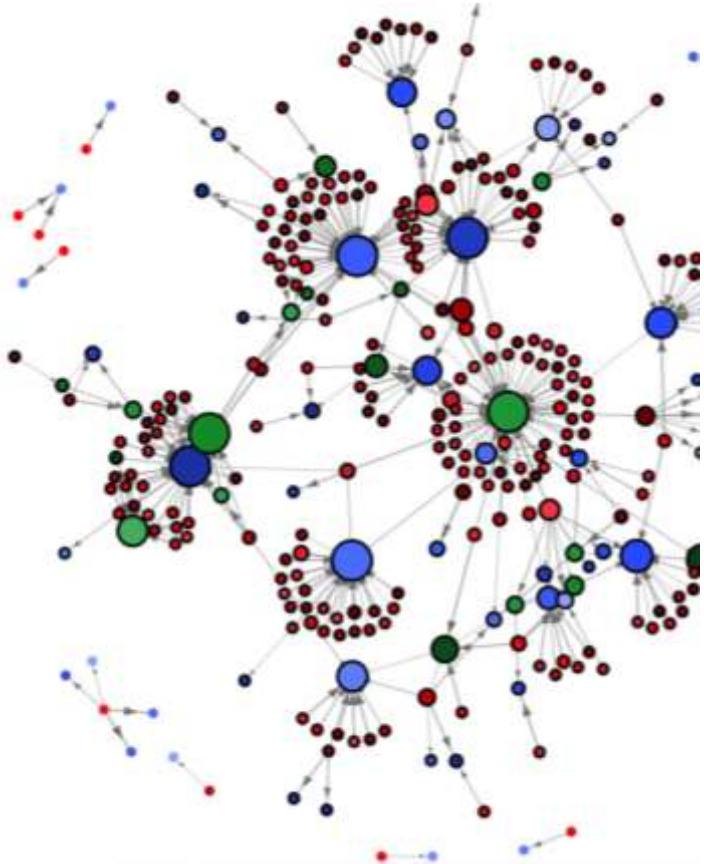


Using ML and AI to ‘inform’ a DevOps factory or pipeline of notable events, usually to help improve the process over time, or help make decisions based on real-time event.

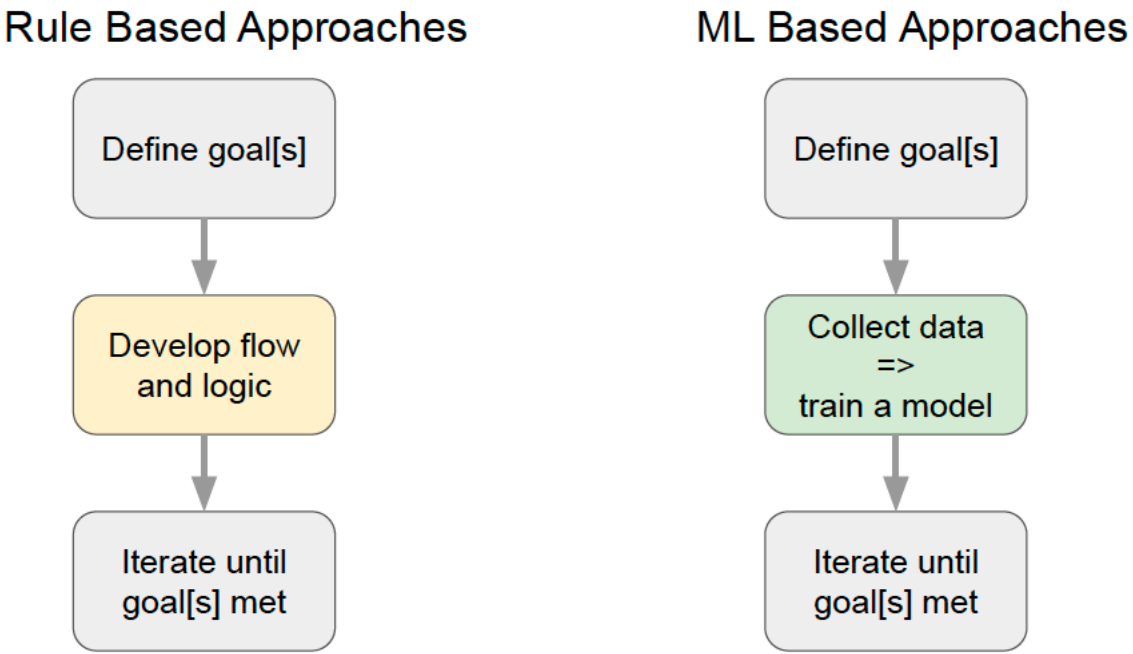
Requirements:

- Monitoring each step
- Must develop models that allow for ‘actionable’ events.

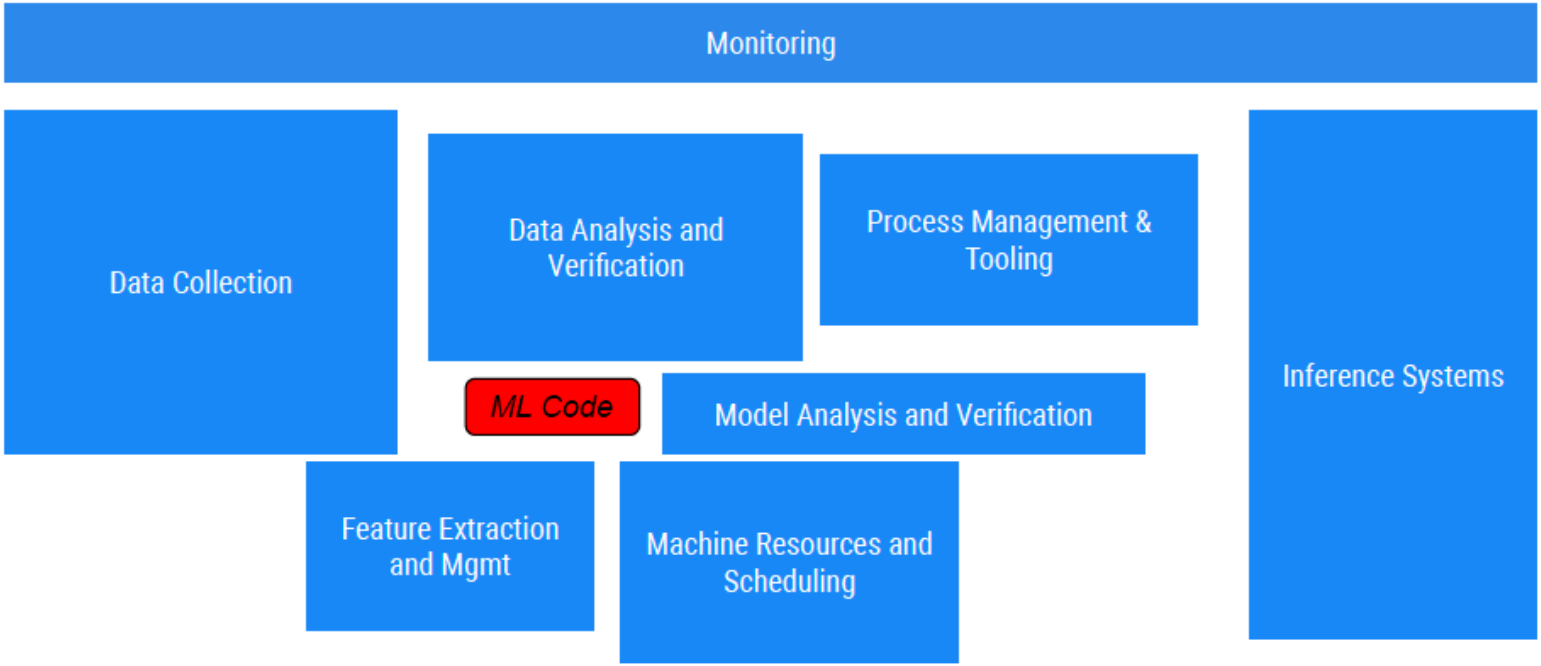
DevOps for AI



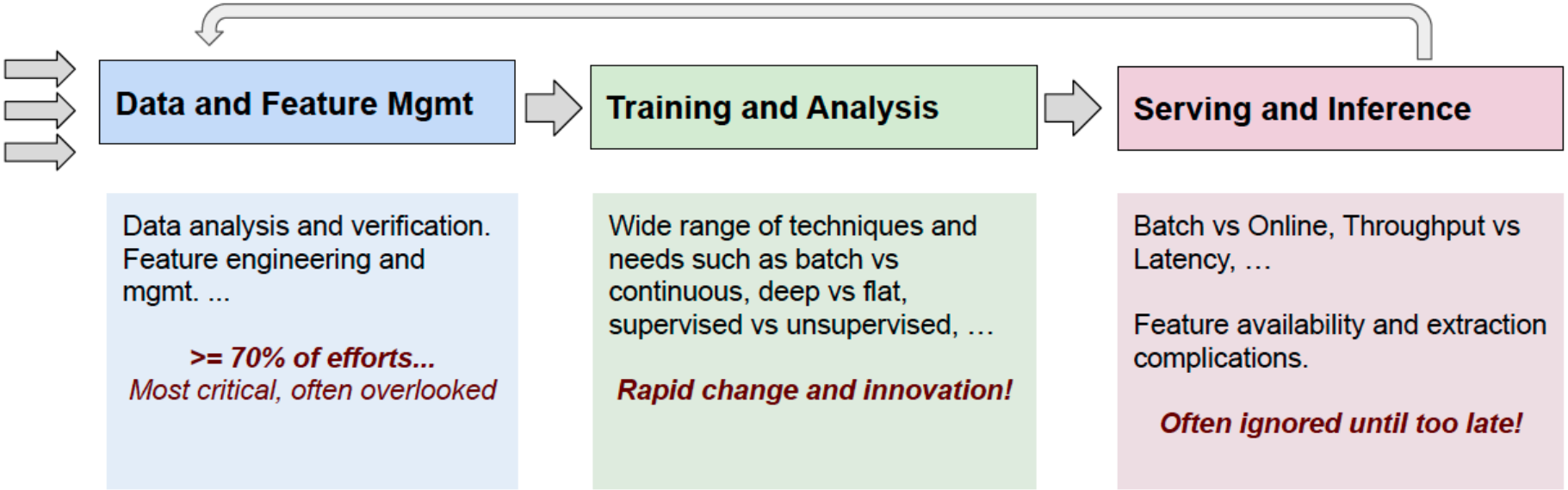
Traditional vs ML Based Approaches?



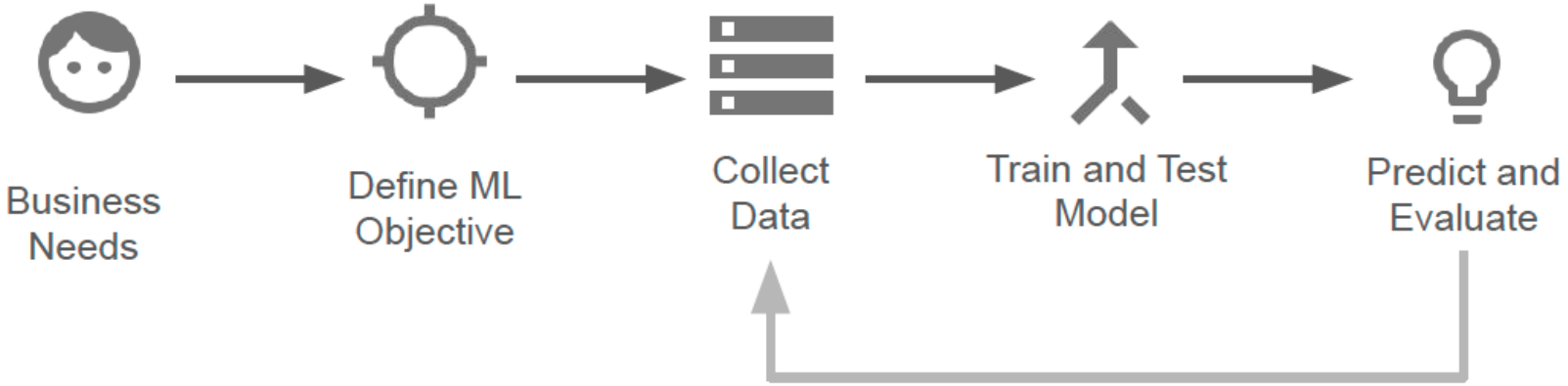
Code in AI System



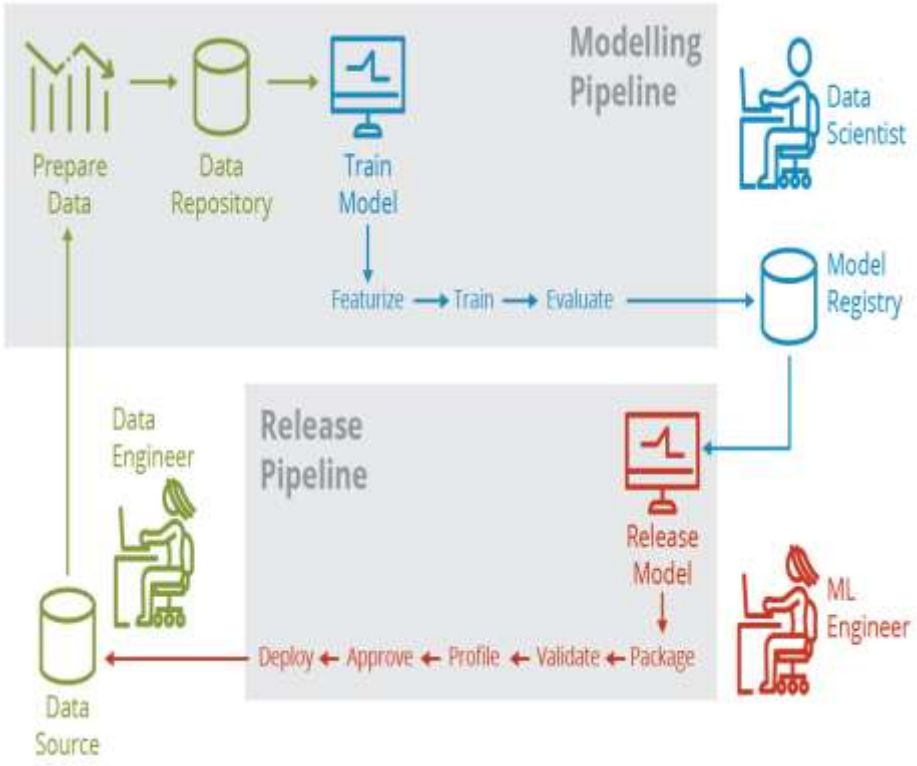
More formally…



ML Development Cycle



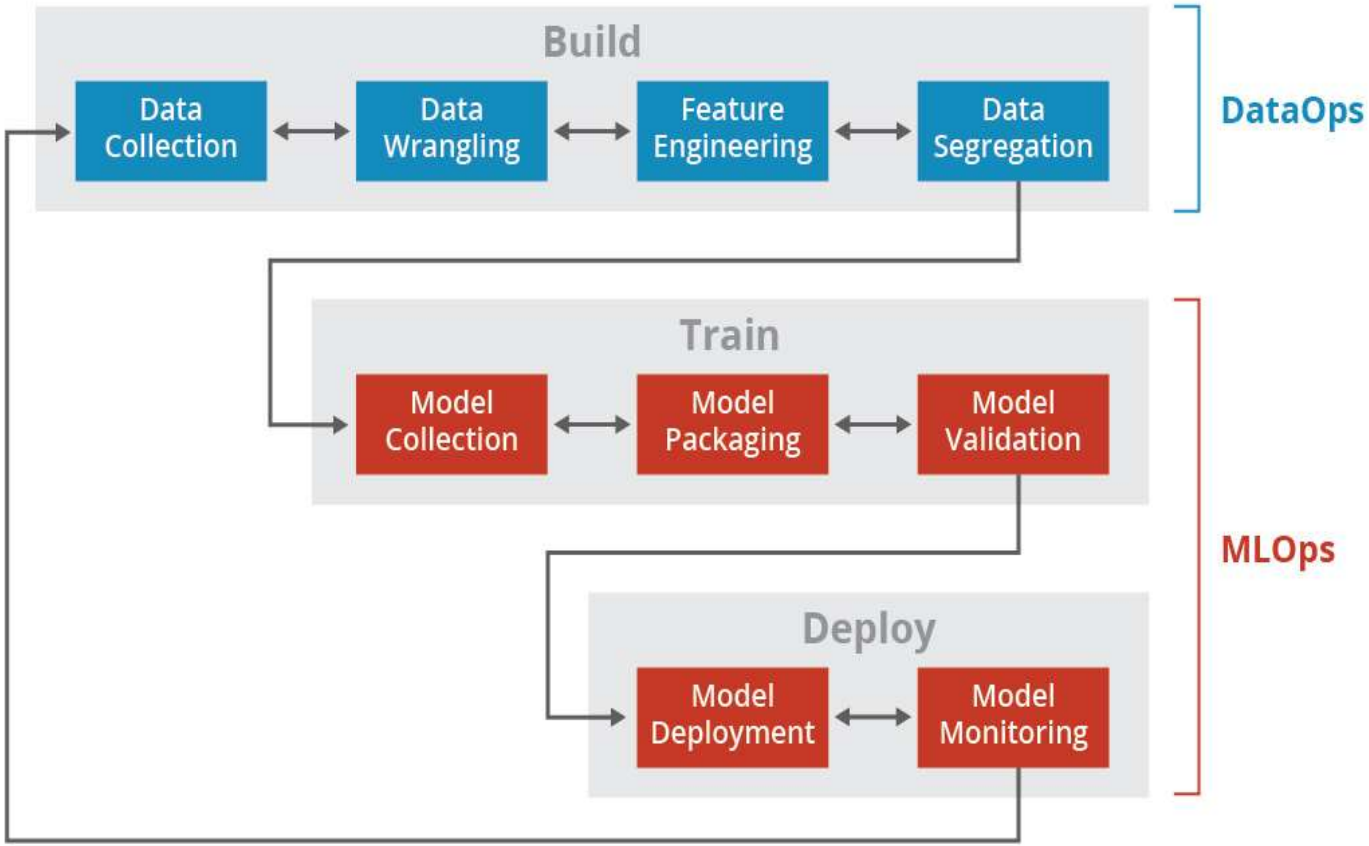
DevOps for AI



Using DevOps concepts and methodologies in every aspect of ML and AI enabled software systems.

- Data curation
- Training data
- Model creation, storage
- Deployment
- Monitoring
- Re-training

DataOps and MLOps exist



Important considerations

- Data must be prepared before model training
- Model release requires operationalization
- Post-deployment monitoring should record all real-world data serving as input to the deployed model
- Team members include
  - Data engineers, Data scientists
  - ML engineers, DevOps
  - Developers
- Model performance
- Deployment strategies
- Model storage and sharing

AI Engineering Framework

AI Technologies and Components

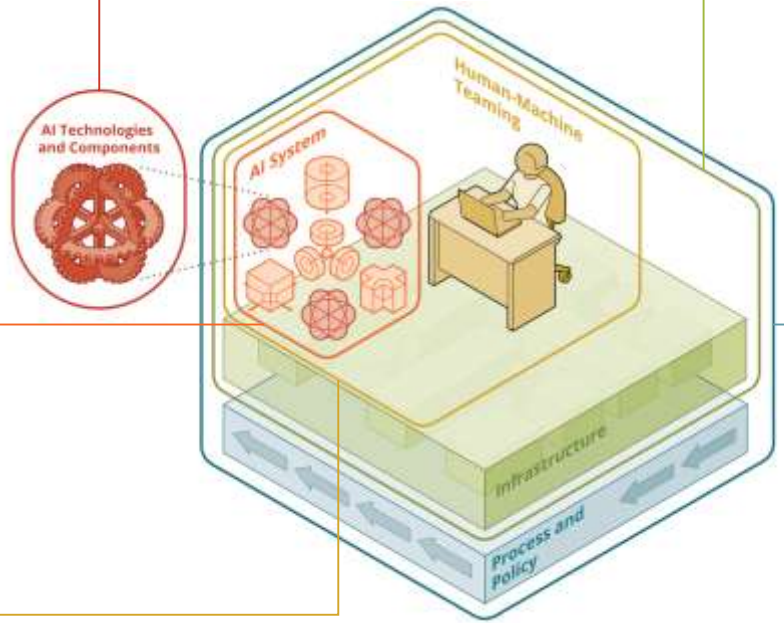
- Knowledge representation
- Modeling and abstraction
- Algorithms
- Scalability and performance
- Robust component design
- Component V&V
- Novelty and Uncertainty

AI System

- Architecture, analysis and design
- Model-based engineering
- Virtual integration
- Robustness and resiliency
- Build security-in Secure AI
- System V&V

Human-Machine Teaming

- Interpretability and explainability
- Human-machine trust
- Machine-human trust
- Co-learning
- Communicating uncertainty
- Data Presentation



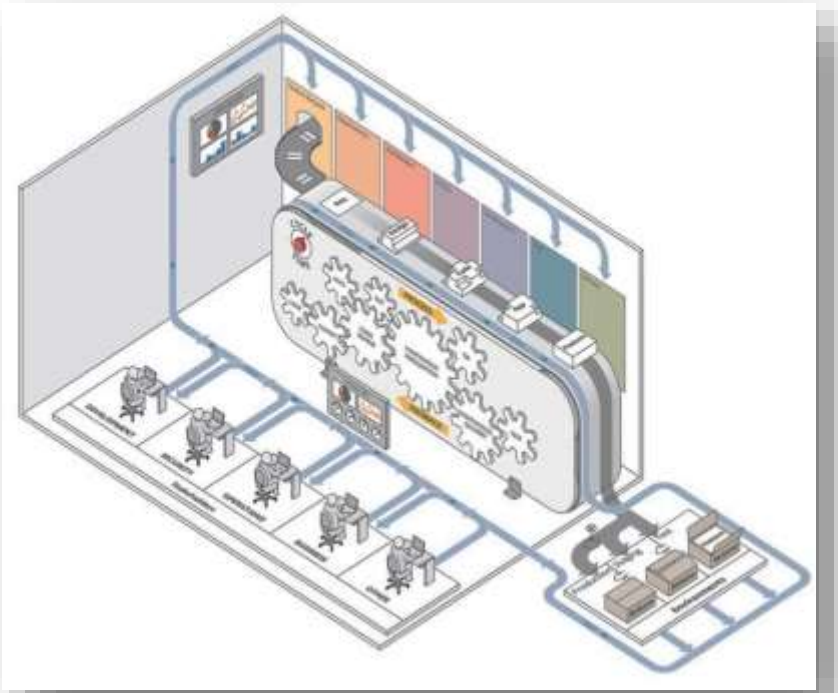
Infrastructure

- Data, data management, and data pipelines
- Scalability, performance, and evaluation
- Computational resources and considerations (CSWaP)
- Processing placement and deployment, data locality
- Platform choices and transferability
- Tooling
- Sensors and actuators
- Future architectures

Process and Policy

- AI system acquisition
- AI-ready modern software practices
- ML lifecycle management
- Data lifecycle management
- Standards
- Benchmarks
- Socio-technical issues
- Economic considerations
- AI-centric Threat Model
- Risk and resiliency
- Security Coordination (AI C/PSIRT)
- Ethics
- Bias
- Privacy

Necessary DevOps Factory additions



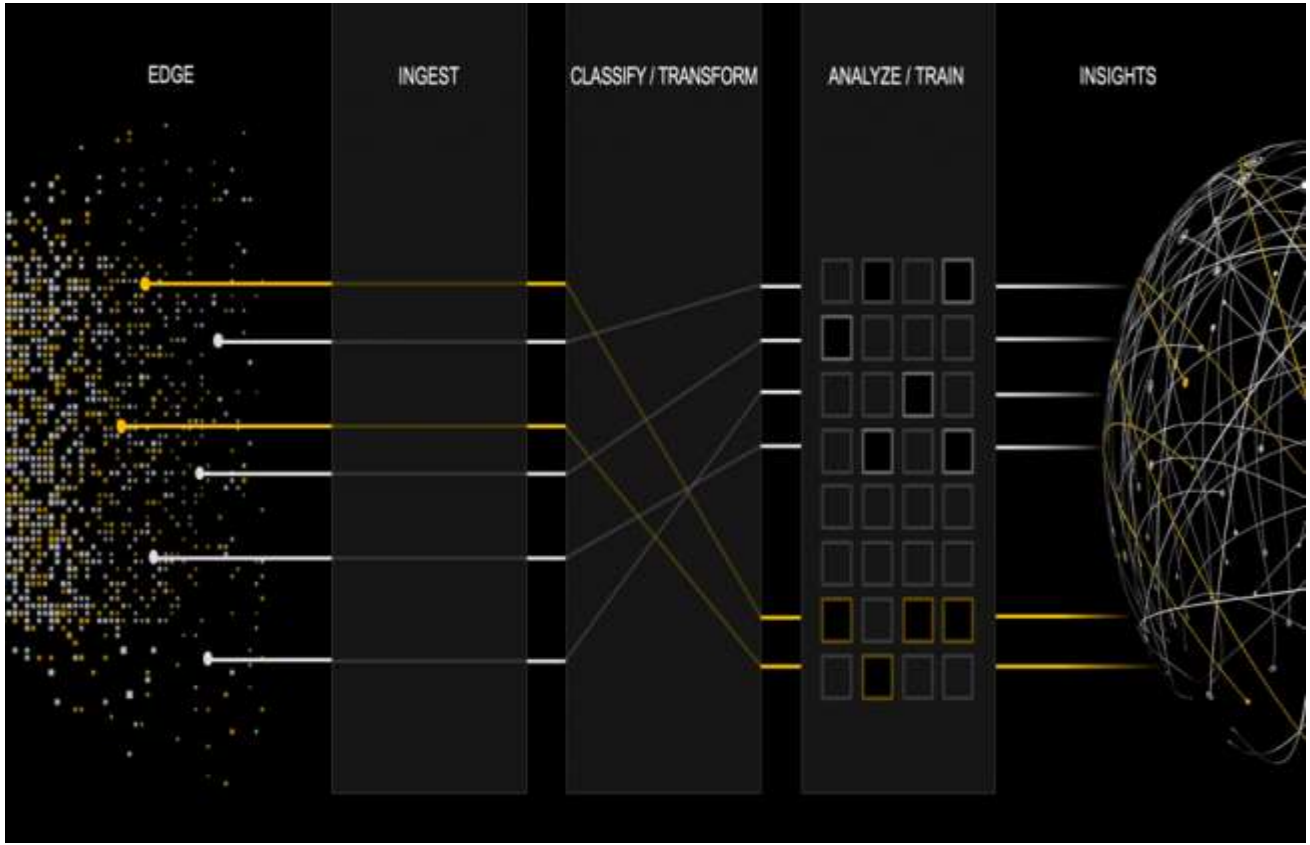
1. Embrace an MLOps culture to facilitate an ML- driven factory
2. Establish a cultural focus on data-driven development to facilitate ML model creation
3. Include data scientists and data engineers in software development teams
4. Automate model deployment via continuous delivery/deployment
5. Establish continuous feedback including model monitoring like *model inputs*, *model outputs and decisions*, *user action and rewards* and *model fairness.*

DevOps for data curation

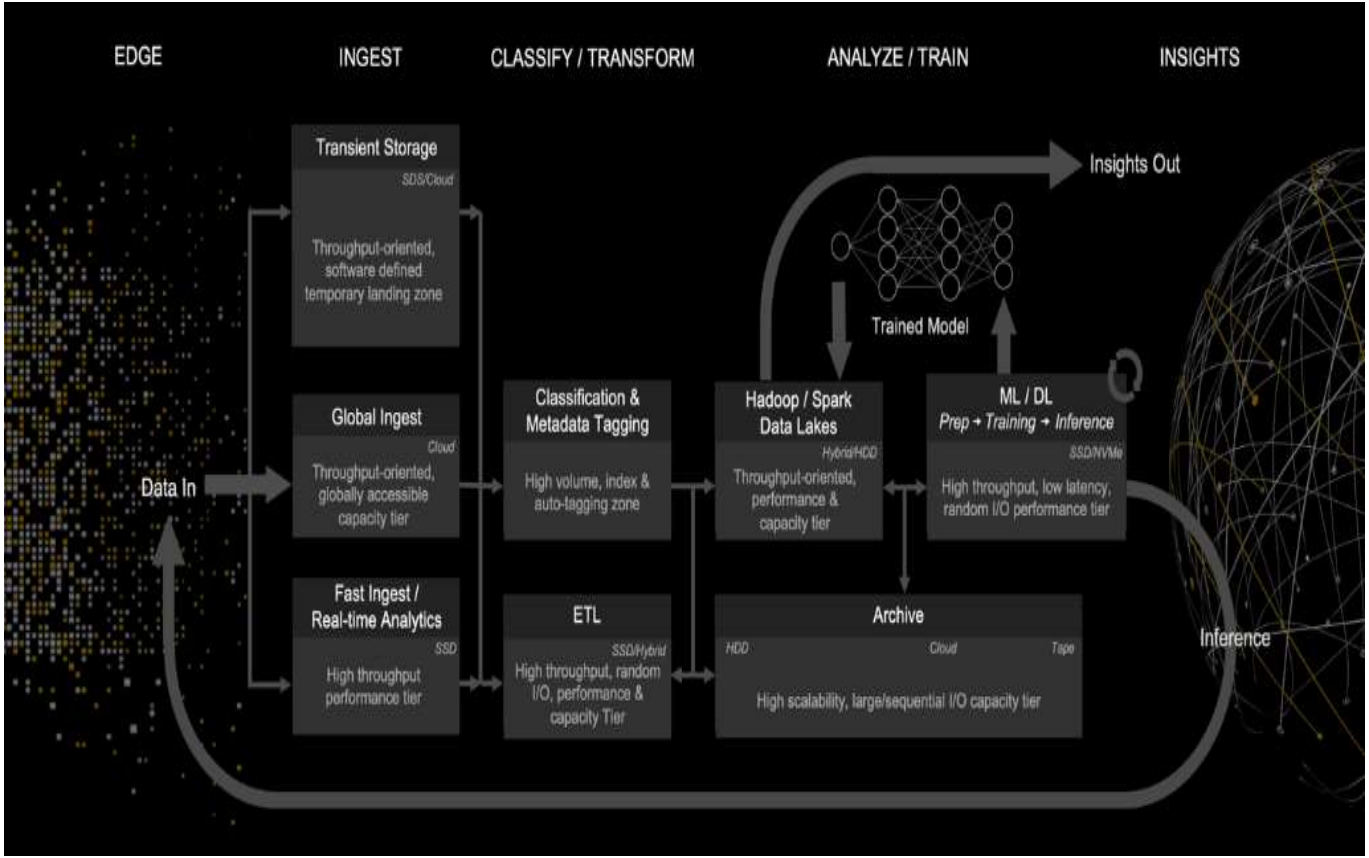
Data is a common critical element of an AI system

- The general process for data processing:
  - Develop business cases
  - Ingest
  - Classify/transform/analyze
  - Insights
  - Availability of Data for DS
  - Validating Data

Data curation



Data curation - workflow



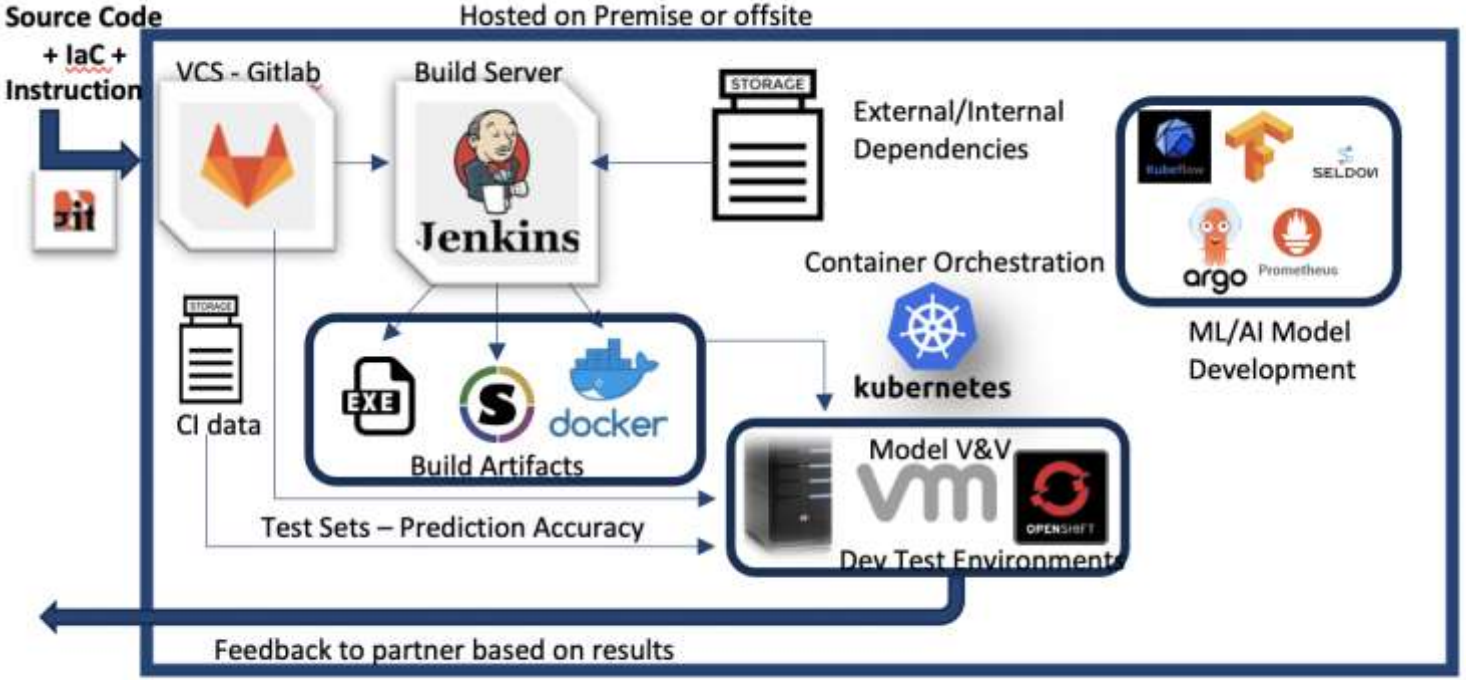
Monitoring deployed AI systems

- Include a return loop of data to the starting point of the pipeline
- Archive all ingress data to the model for future training
- Record and analyze the model's output for functionality and integrity
- Determine if a model requires modification or re-training
- Model inputs: what data, predictions or recommendations
- Model outputs and decisions
- Model interpretability outputs
- Example: using EFK stack for monitoring and observability
  - Elasticsearch: an open source search engine.
  - FluentD: an open source data collector for unified logging layer.
  - Kibana: an open source web UI that makes it easy to explore and visualize the data indexed by Elasticsearch.

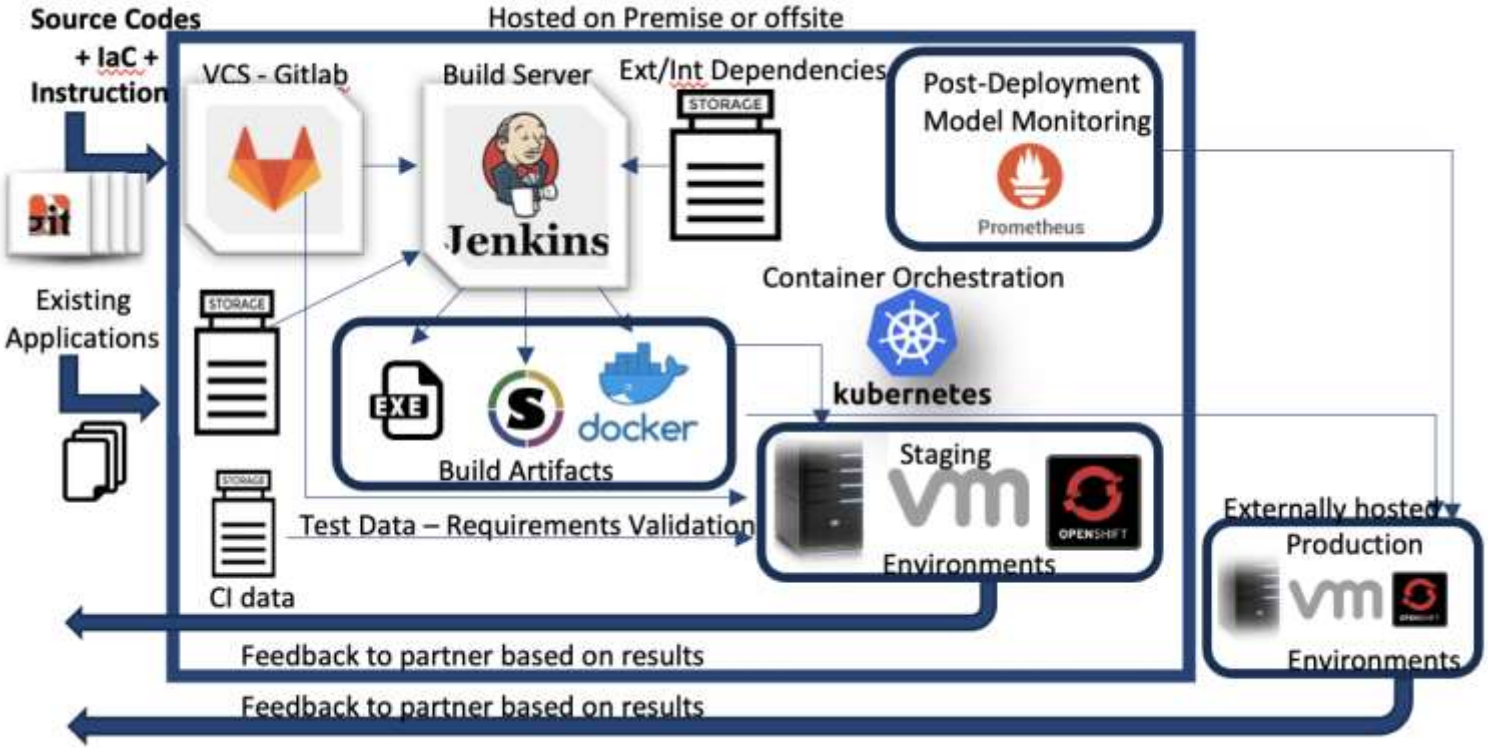
Additional guidance for an AI/ML Pipeline

- Capable of ingesting multiple data types
- Data maintained and versioned
  - Data Version Control (dvc.org)
- Real-time monitoring
- Responsive to changing conditions discovered during monitoring
- Traceability
- Language standardization

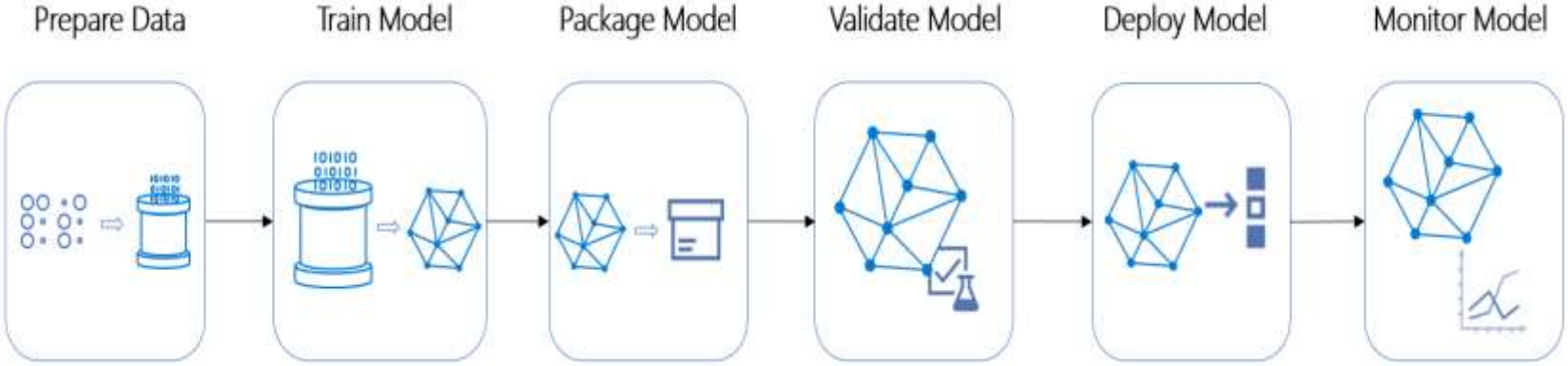
Exemplary AI+DevSecOps implementation scenarios



Exemplary AI+DevSecOps implementation scenarios



Summary



- Use DevOps to build, deploy, and monitor systems so that a pathway exists to take action on a ML/AI enabled system.
- These ‘actions’ could improve model performance, system security, and many other possibilities

For more information...

DevOps: https://www.sei.cmu.edu/go/devops

DevOps Blog: https://insights.sei.cmu.edu/devops

Webinar : https://www.sei.cmu.edu/publications/webinars/index.cfm

Podcast : https://www.sei.cmu.edu/publications/podcasts/index.cfm

Thank You

Hasan Yasar
Technical Director, Adjunct Faculty Member
Continuous Deployment of Capability
hyasar@sei.cmu.edu
@securelifecycle