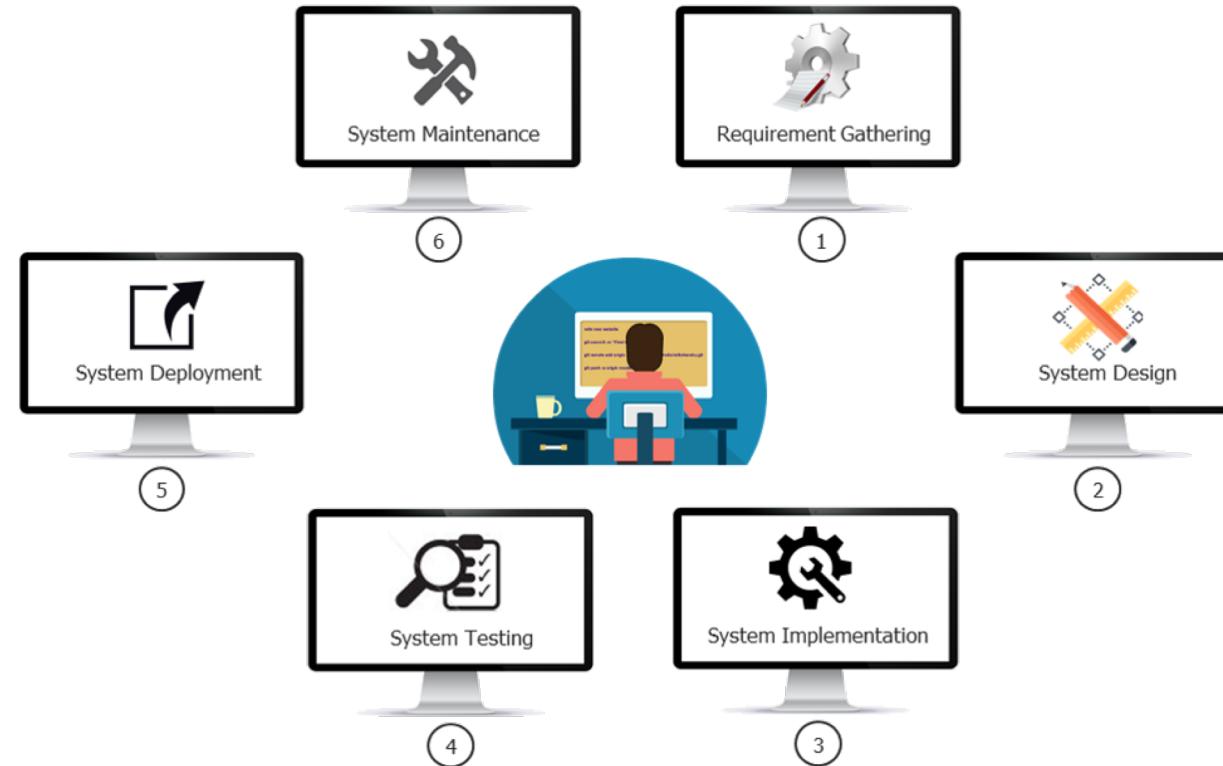


# DevOps



# Traditional Development Models: Waterfall

**Waterfall Model:** New phase in the development process begins only if the previous phase is complete





# What is DevOps?

- DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity
- DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software
- This speed enables organizations to better serve their customers and compete more effectively in the market.



# DevOps

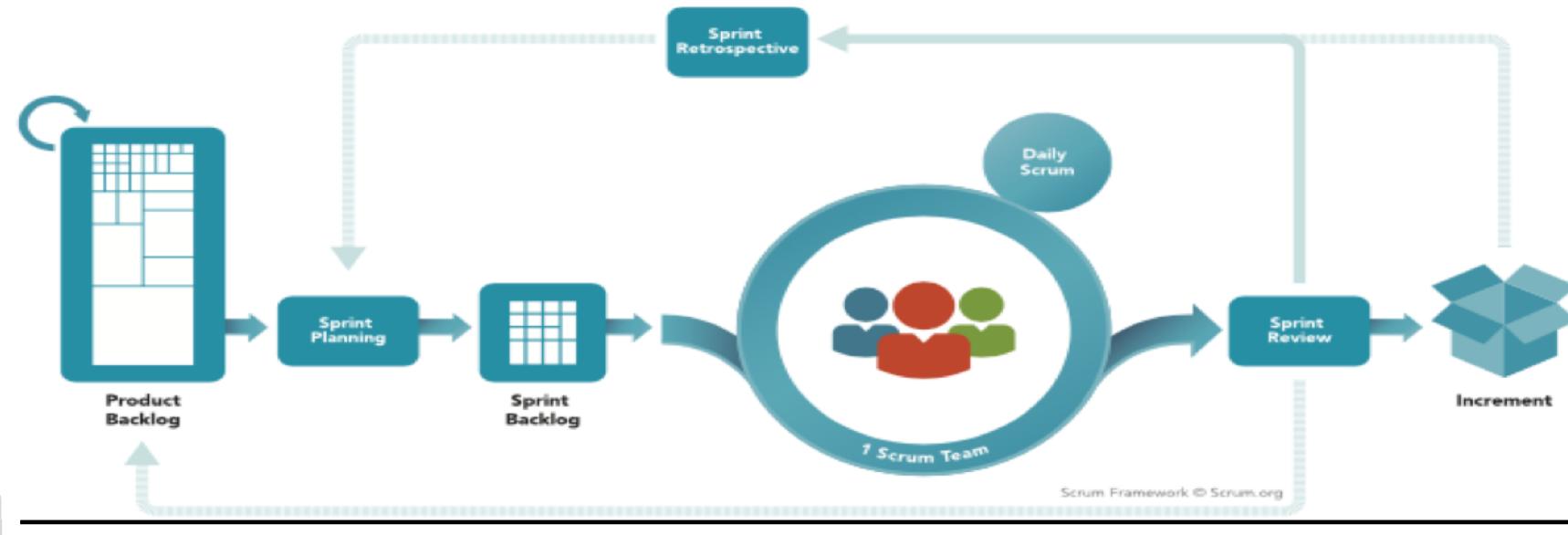
- DevOps is short for Development Operations
- It focuses on collaboration between developers and other parties involved in building, deploying, operating, and maintaining software systems
- It is about defining a flow from development through full scale operation of a system
- It is about Systems Thinking with feedback to earlier stages of a DevOps workflow
- The emphasis is on automating processes required to release and change software
  - With a view to rapid, frequent, and reliable releases



# Agile

- Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto
- Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context

- From agilealliance.org





# Agile & DevOps

- Some of the principles of Agile manifesto that lead to a DevOps environment
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
  - Business people and developers must work together daily throughout the project
- DevOps not only talks about collaboration between Development and Operations, but improved collaboration between all teams including business
- DevOps also promotes using a lot of automation to complete the many tasks and steps that go into making a deliverable potentially shippable increment
- Using tools to replace the non-human steps like build creation, environment setup, trigger test cases, automated deployments multiple times in a day



# How DevOps Works?

- As part of devops practice Development and operations teams are no longer “siloed.”
- Typically a single team will work on the entire application lifecycle, from development and test to deployment to operations
- Quality assurance and security teams also become more tightly integrated with development and operations and throughout the application lifecycle
- When security is the focus of everyone on a DevOps team, this is sometimes referred to as **DevSecOps**



# DevOps Practices

- Continuous Development and Testing : perform very frequent but small updates
- Continuous Integration (CI) with Pipeline as code
- Continuous Delivery
- Continuous Deployment (CD)
- Continuous Monitoring
- Infrastructure Automation with Infrastructure As Code
- GitOps

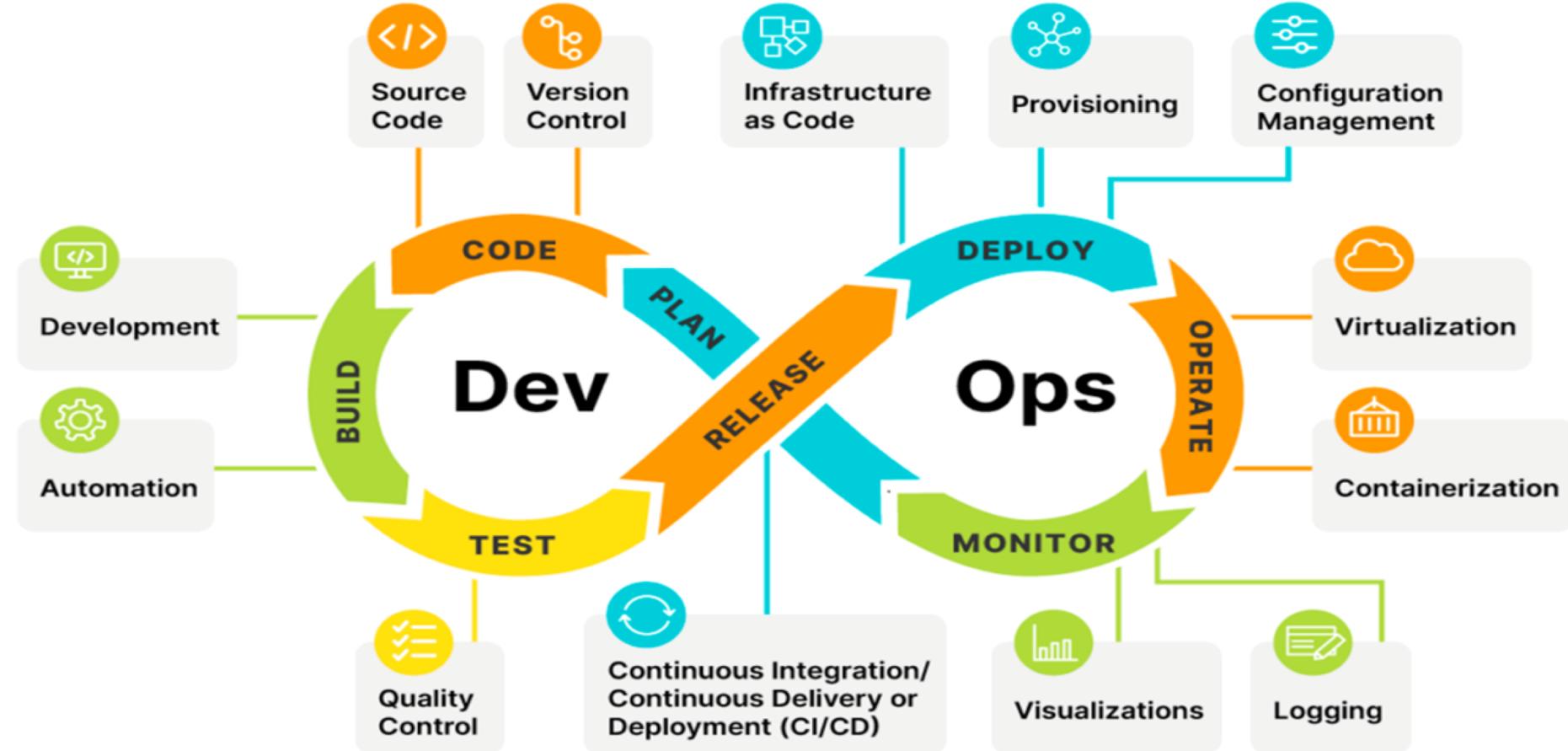


# DevOps Benefits

- Faster, better product delivery
- Improved Collaboration
- Issue identification Early in the lifecycle
- Faster issue resolution and reduced complexity
- Greater scalability and availability
- More stable operating environments
- Better resource utilization
- Greater automation
- Greater visibility into system outcomes
- Greater innovation



# DevOps Journey

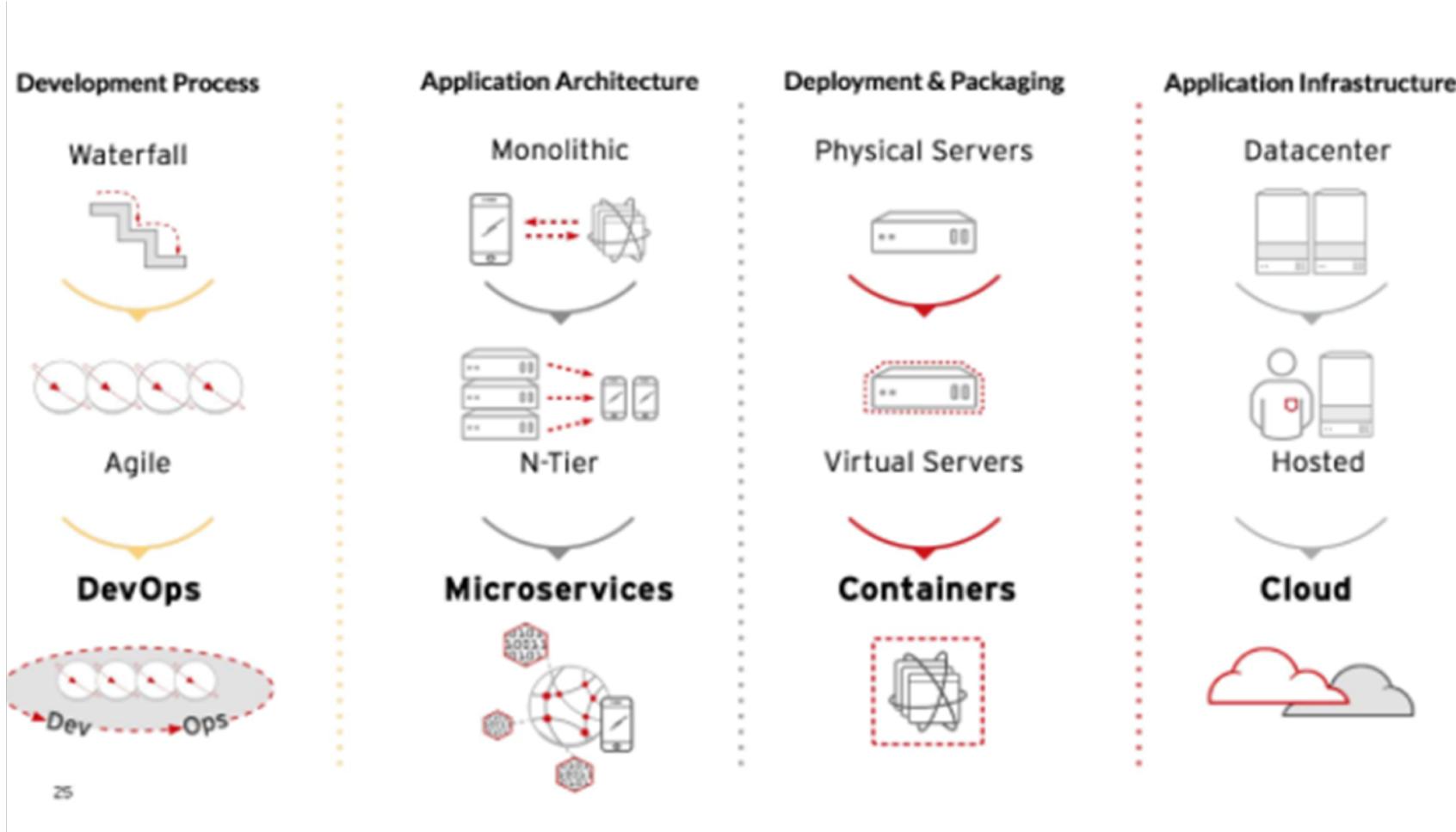




# What DevOps is Not

- A title
- A separate team
- A tool
- Only culture
- Only automation
- Anarchy
- A one size fits all strategy

# Evolution of IT





# Why DevOps

- Faster Time to Market
- Improved Quality
- Enhanced Collaboration
- Increased Stability and Reliability
- Scalability
- Cost Efficiency
- Continuous Feedback and Improvement
- Security Integration
- Agility and Flexibility
- Competitive Advantage



# Cross Functional Teams

- Diverse Expertise
- Common Goal
- Collaboration
- Shared Responsibility
- Faster Problem Solving
- Innovation
- Flexibility
- Accountability
- Project Management
- Continuous Improvement



# DevOps goals/Benefits

- Faster Time to Market
- CI & CD
- Improved Collaboration
- Enhanced Quality and Reliability
- Automation
- Scalability and Resource Efficiency
- Security Integration (DevSecOps)
- Continuous Monitoring and Feedback
- Infrastructure as Code
- Cost Efficiency
- Agile and Lean Principles
- Faster Recovery from Failures
- Cultural Transformation
- Competitive Advantage
- Improved Compliance and Governance



# DevOps Goals

- Smaller, more frequent releases
- Reduced effort and risks
- Reduced cost of product iterations and delays
- A culture of communication and collaboration
- Consistency and speed through automation



# DevOps Stakeholders

- Architects,
- Business representatives,
- Customers,
- Product Managers,
- Project Managers,
- Quality - QA,QC
- Suppliers

Dev



- Systems Engineers,
- Systems Administrators
- Information security,,
- IT Operation engineers,
- Release engineers,
- Database administrators,
- Network engineers,
- Support Professionals,
- Suppliers

Ops





# Benefits of DevOps

- A sustainable, successful business is more than the development and operations teams
- Shift focus from what to why – question, innovate, collaborate
- Repeatable, reliable, and predictable
- Faster reaction to business need
- Higher return on investment in software
- DevOps principles can also be applied to the business



# What's Driving DevOps

## DEPLOYMENTS AT AMAZON.COM

11.6s

1,079

10,000

30,000

Mean time between  
deployments  
(weekday)

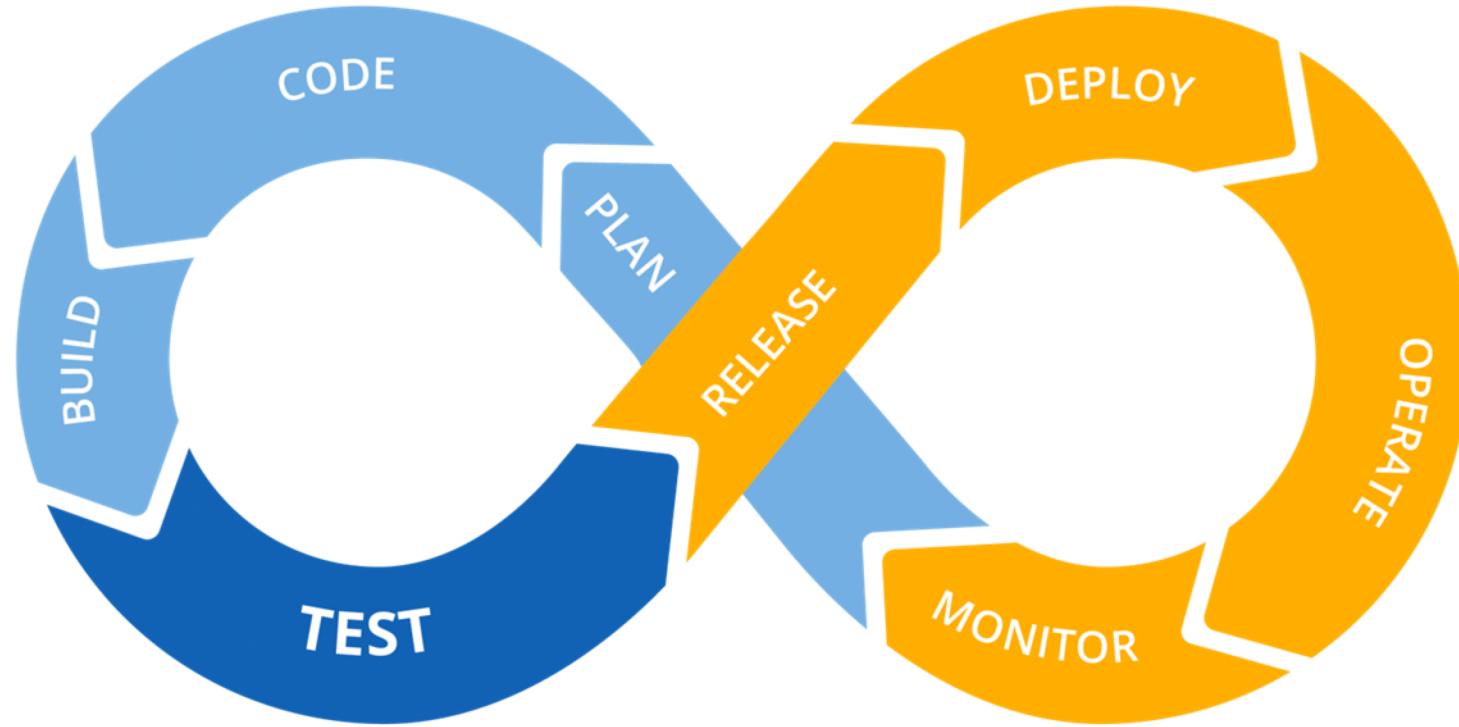
Max number of  
deployments in a  
single hour

Mean number of  
hosts  
simultaneously  
receiving a  
deployment

Max number of  
hosts  
simultaneously  
receiving a  
deployment

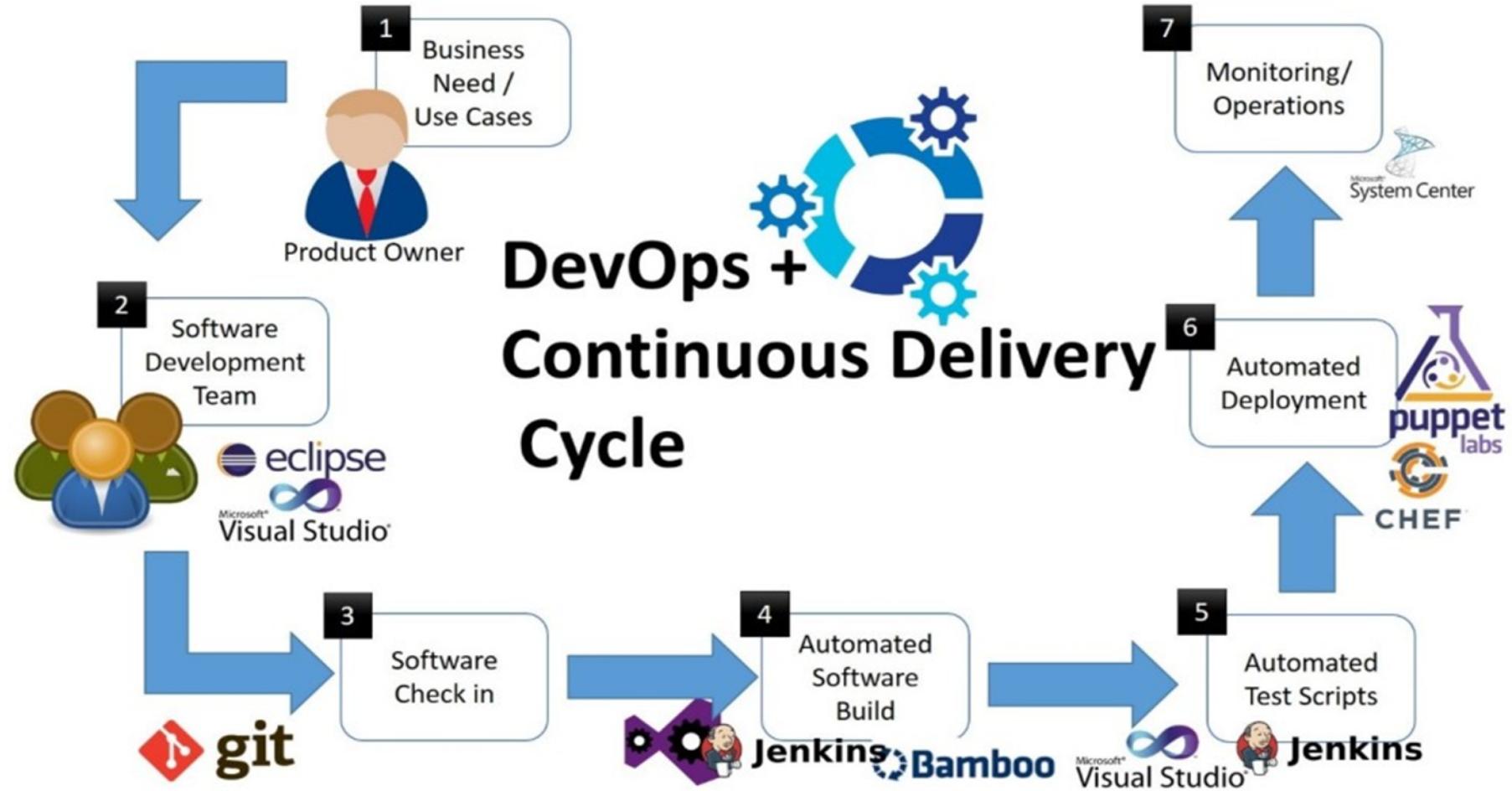


# DevOps Lifecycle





# DevOps Workflow



# Modern DevOps

Learning Path



# Modern DevOps Path

- Base Commands
- Branches
- Merges
- Pull Request

## 1. Git

## 2. Programming Language

- Python
- Go
- JavaScript
- Java

- Shell Commands
- File System
- Networking
- Virtualization

## 3. Linux



# Modern DevOps Path

- OSI Model
- DNS
- HTTP
- SSH
- Security Tools

## 4. N/W Security

## 5. Server Management

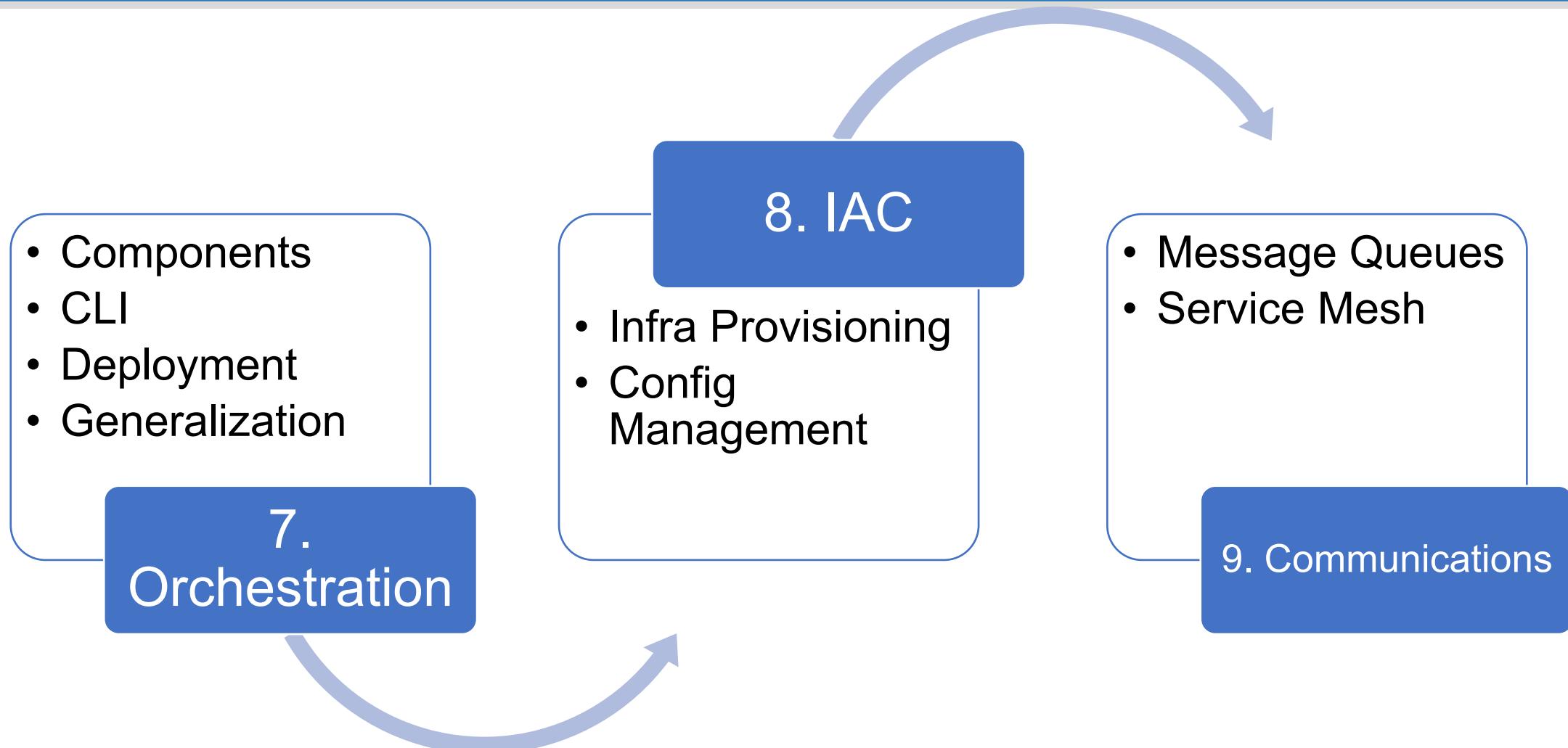
- Reverse proxy
- Caching Server
- Firewall
- Web Server
- API Gateways
- Load Balancers

- Containerization
- Docker Files
- Networking
- Docker Compose

## 6. Containers



# Modern DevOps Path





# Modern DevOps Path

- Setup Server
- Integration
- Pipeline Tools

## 10. CI/CD

## 11. Monitor & Observability

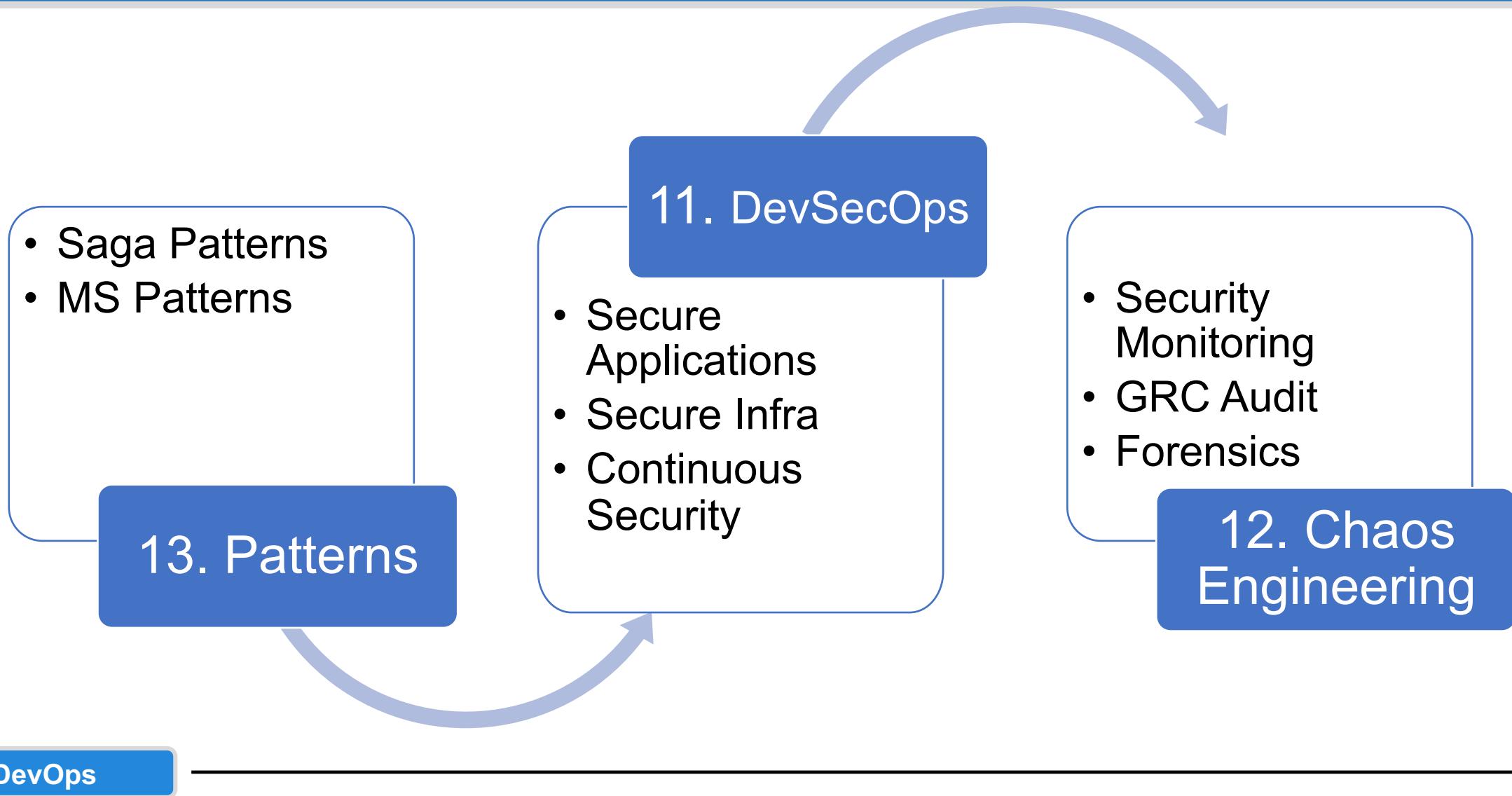
- Infra Monitoring
- Application Monitoring
- Logs Management
- Open Telemetry

- SCRUM
- Built In Quality
- Troubleshooting

## 12. Principles & Practices



# Modern DevOps Path



# Thank You