How leaders set the stage - successfully scaling DevSecOps

Tim Anderson
Sr. Security Advisor,
AWS Security
tdander@amazon.com

Outcomes

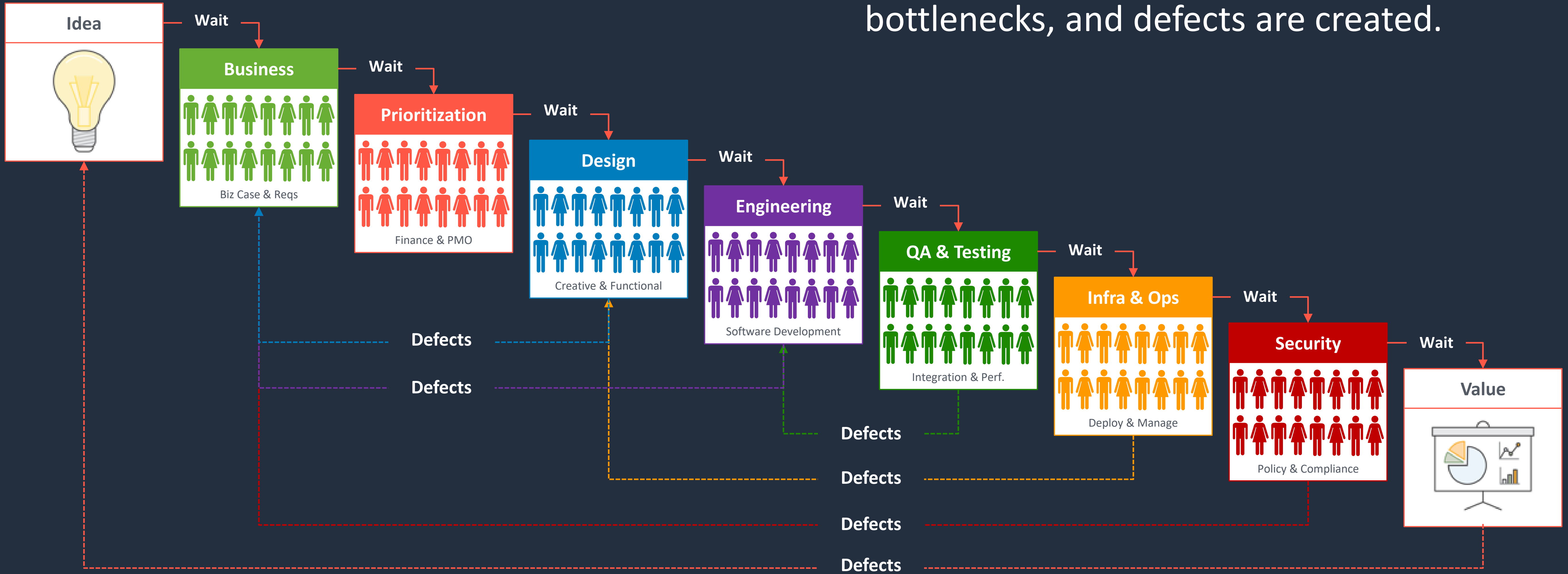
1. A strategy for scaling adoption

2. Mechanisms to build security at scale

DevSecOps strategy that scales

Innovation drain

In the process, pervasive handoffs, bottlenecks, and defects are created.



Defects passed downstream are often discovered late in the delivery cycle and have to be revisited.

Each Step Delays Time-to-Value

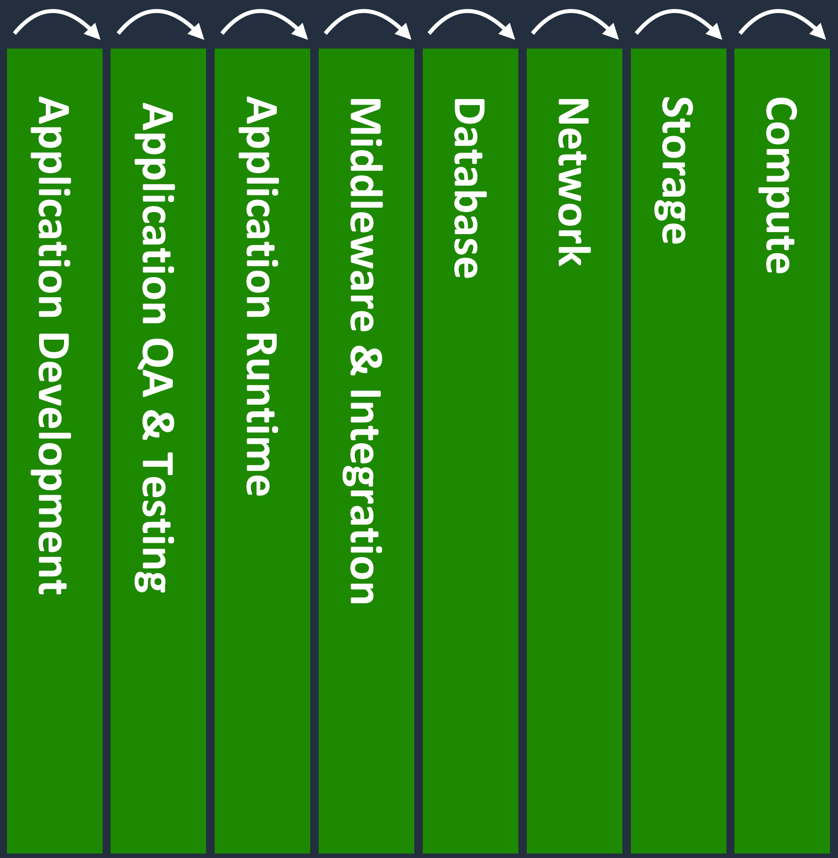
Traditional IT vs. Full Stack Engineering

Challenges

- Handoffs
- Resource Constraints
- Competing Priorities
- Wait Time

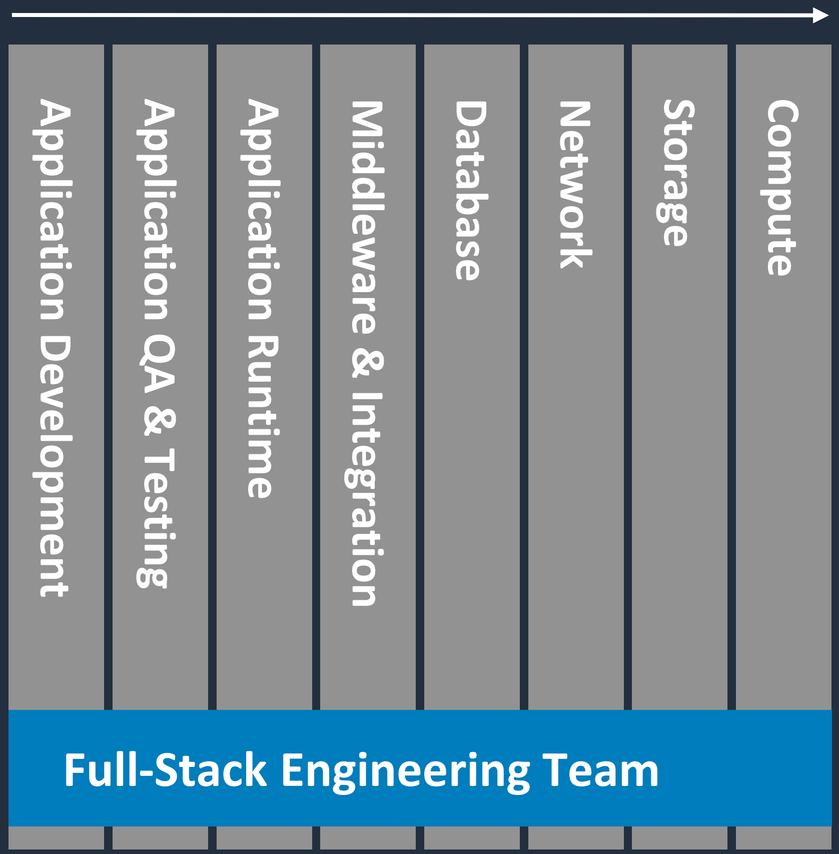
Traditional IT

activity-based teams



Full-Stack Engineering

outcome-based teams



Considerations

- Infrastructure
- Operations
- Security
- Finance

The Benefits

- Fast time to market or time to value
- Lower costs
- Less waste in processes
- Reduced risk
- Increased innovation
- Better operational controls through automation

Tenets of DevSecOps

1. Everyone is a security owner
2. Test security as early as possible to accelerate feedback.
3. Prioritize preventive security controls to stop bad things from happening.
4. When deploying a detective security control, ensure it has a complementary responsive security control to do something about it.
5. Automate, automate, automate.

Driving Change - Area of Focus

FROM	TO
HiPPO-based decision-making	Data-driven decisions that are tested and measured
Large feature sets and systems sprawl	Constantly re-prioritizing and validating for relevance
Protecting the core business	Continuous refactoring and improvement
Business and IT silos	Teams that span business and technology
Big bets that languish	Reduced batch size and frequency of releases
Software and processes that aren’t nimble	Reducing the lead time from idea to implementation
Planning for best case operating state	Assuming attack and failure
Gated opaque security slows the business	Security as quality - business driver and differentiator

Be aware of top 5 pitfalls

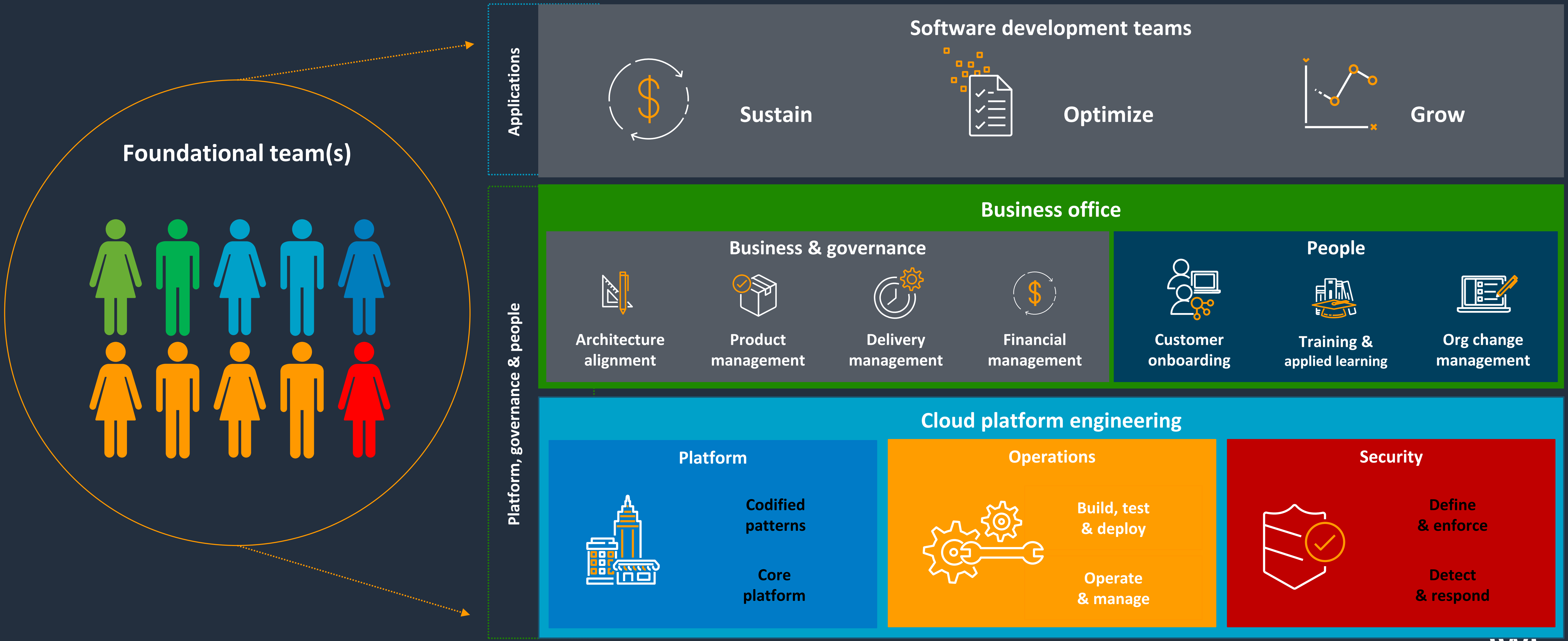
1. Lack of Executive Sponsorship
2. Poor Communications
3. Insufficient Resource Allocation
4. Undefined KPI’s and Outcomes
5. Workforce Management

How the team drives change

- Building reusable patterns / Product focused
- Ingraining security with every team member
- Visibility of team operations
- Continuous improvement – feedback cycle and actions
- Look to simplify

How do you start?

Think big, but start small. Launch a cloud foundation team and a small number of development teams to start the flywheel. Scale as the transformation accelerates and expands.



Foundational team(s)

Applications

Software development teams

Sustain

Optimize

Grow

Platform, governance & people

Business office

Business & governance

Architecture alignment

Product management

Delivery management

Financial management

People

Customer onboarding

Training & applied learning

Org change management

Cloud platform engineering

Platform

Codified patterns

Core platform

Operations

Build, test & deploy

Operate & manage

Security

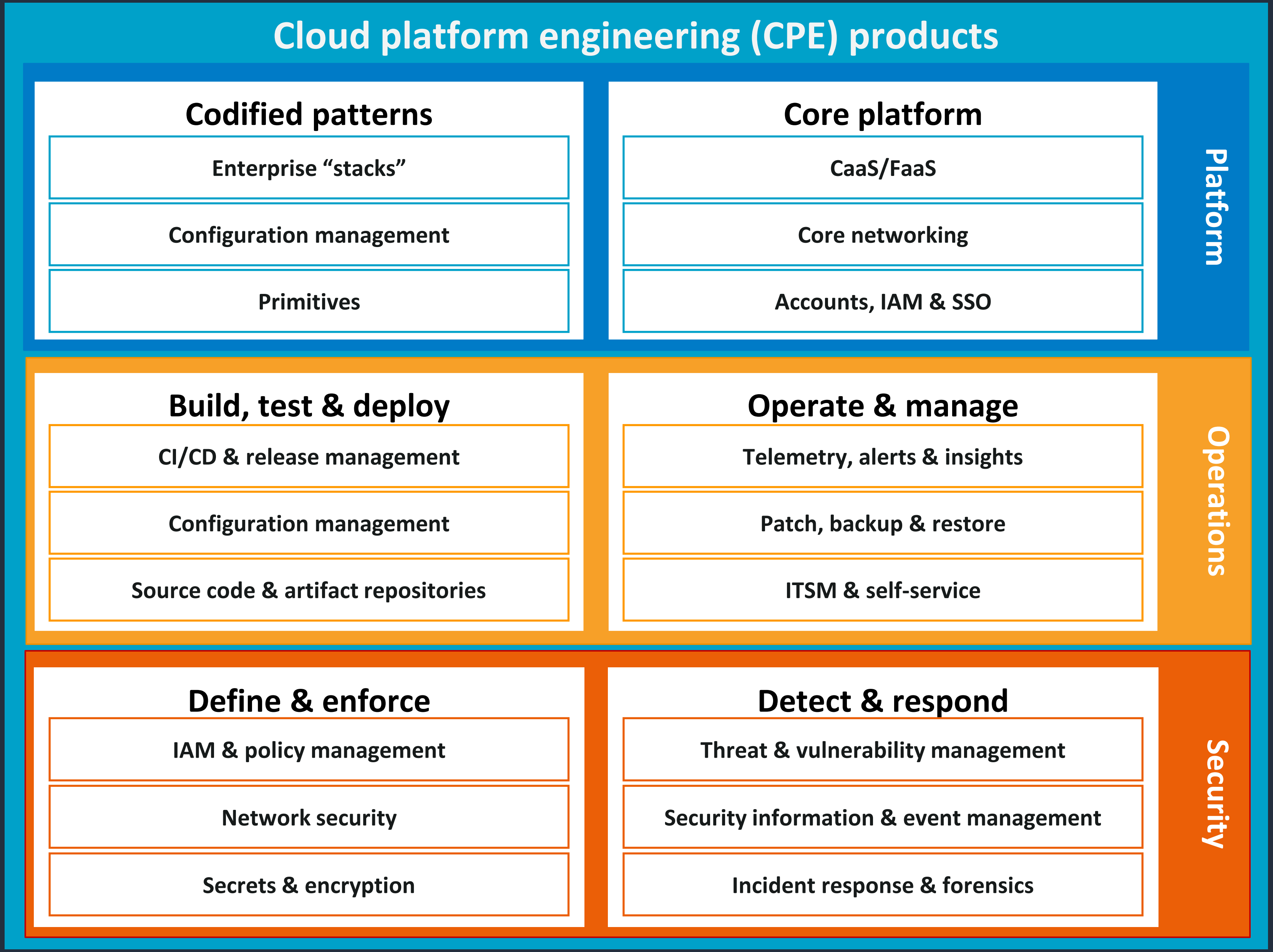
Define & enforce

Detect & respond

What products does cloud platform engineering provide?

Cloud platform engineering

Codifies differences between stock AWS service configurations and the enterprise's standards, packaged and continuously improved as self-service deployable products to customers



Cloud platform engineering (CPE) products

Platform

Codified patterns
- Enterprise "stacks"
- Configuration management
- Primitives

Core platform
- CaaS/FaaS
- Core networking
- Accounts, IAM & SSO

Operations

Build, test & deploy
- CI/CD & release management
- Configuration management
- Source code & artifact repositories

Operate & manage
- Telemetry, alerts & insights
- Patch, backup & restore
- ITSM & self-service

Security

Define & enforce
- IAM & policy management
- Network security
- Secrets & encryption

Detect & respond
- Threat & vulnerability management
- Security information & event management
- Incident response & forensics

Critical Success Factors for Successful Transformation

Visible and committed leadership *("management driving the change")*	**Targeted and effective communications** *("adapting the communication strategy")*
Compelling need for change *("establishing a high enough sense of urgency")*	**Single program focus** *("prioritizing projects and allocating resources")*
Clarity of direction *("grounding the vision of the desired state")*	**Measurable goals** *("setting reachable milestones")*
Broad-based participation *(“engage key impacted audiences")*	**Disciplined project management** *("running the project effectively")*

Security mechanisms for DevSecOps

Organizational change

- Move Security up the value chain
- Security as quality
- Lead communities of practice
- Ensure cloud awareness
- Not a team of "no"

Giving security confidence – Proving Assurance

Threat modeling

Feed security cases to the Dev team - work it like high priority defects

Address separation of duties concerns

Adopting zero known defect approach

Continuously vet/audit security in dev and prod

- Rigorous testing in each environment
- Peer review - Each technologist should be thinking about possible defects and possible security vulnerabilities. Code should always be reviewed by a peer, who should also be looking for vulnerabilities

General best practices

CI/CD is a MUST!

Clean room

Everything into a repository

Start with continuous delivery

Deploy to staging, test, deploy to an AZ, test, deploy to a Region, test

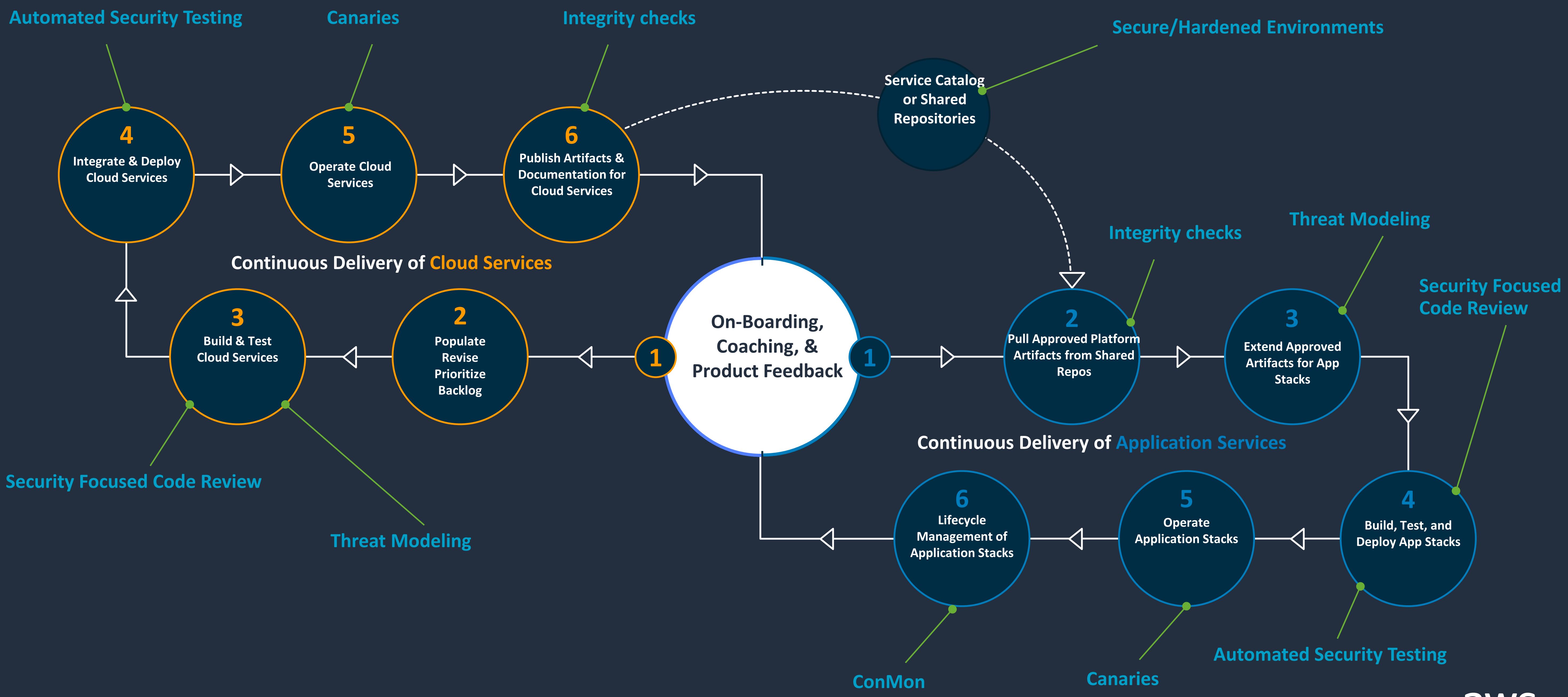
Code Reviews are one of the best mechanisms for “good” code

Style checkers

Auto-rollbacks

Meaningful dashboards

Team Interaction and Workflow


Automated Security Testing
Canaries
Integrity checks
Secure/Hardened Environments
4
Integrate & Deploy Cloud Services
5
Operate Cloud Services
6
Publish Artifacts & Documentation for Cloud Services
Service Catalog or Shared Repositories
Continuous Delivery of Cloud Services
Integrity checks
Threat Modeling
Security Focused Code Review
3
Build & Test Cloud Services
2
Populate Revise Prioritize Backlog
1
On-Boarding, Coaching, & Product Feedback
1
2
Pull Approved Platform Artifacts from Shared Repos
3
Extend Approved Artifacts for App Stacks
Security Focused Code Review
Threat Modeling
Continuous Delivery of Application Services
6
Lifecycle Management of Application Stacks
5
Operate Application Stacks
4
Build, Test, and Deploy App Stacks
ConMon
Canaries
Automated Security Testing

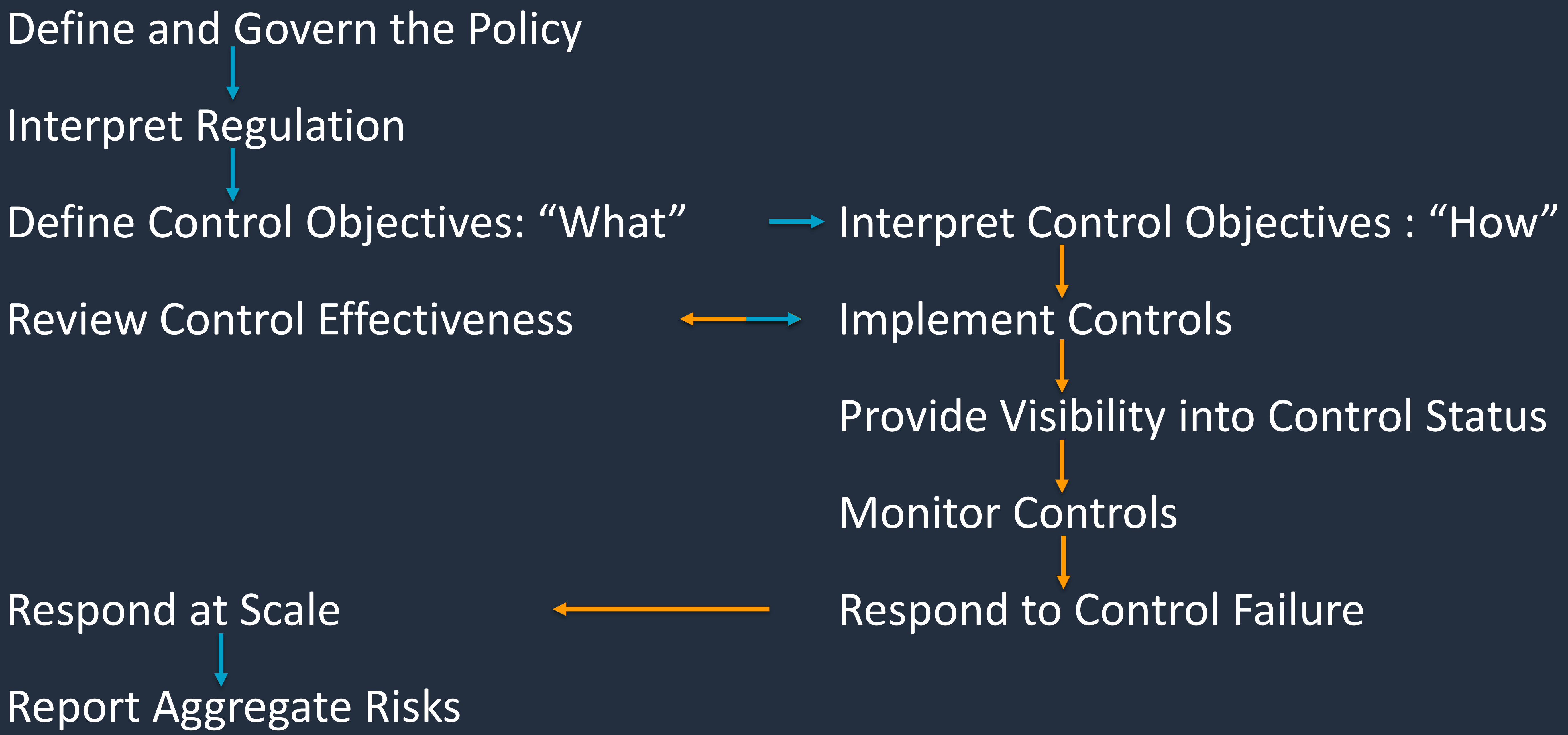
Consistency Breeds Trust

CI/CD

- Deeply understand your SDLC
- Catalog the controls
- Document every instance of human interaction
- Reduce human access
- Set a goal to deploy workloads from source.

Security Org

Product/Service Teams



Define and Govern the Policy

Interpret Regulation

Define Control Objectives: “What”

Review Control Effectiveness

Respond at Scale

Report Aggregate Risks

Interpret Control Objectives : “How”

Implement Controls

Provide Visibility into Control Status

Monitor Controls

Respond to Control Failure

Thank You!

Tim Anderson
Sr. Security Advisor,
AWS Security
tdander@amazon.com