We’re Sorry.
Love, DevOps

Dear Security, Compliance, and Auditors

Bill Bensing
Managing Architect

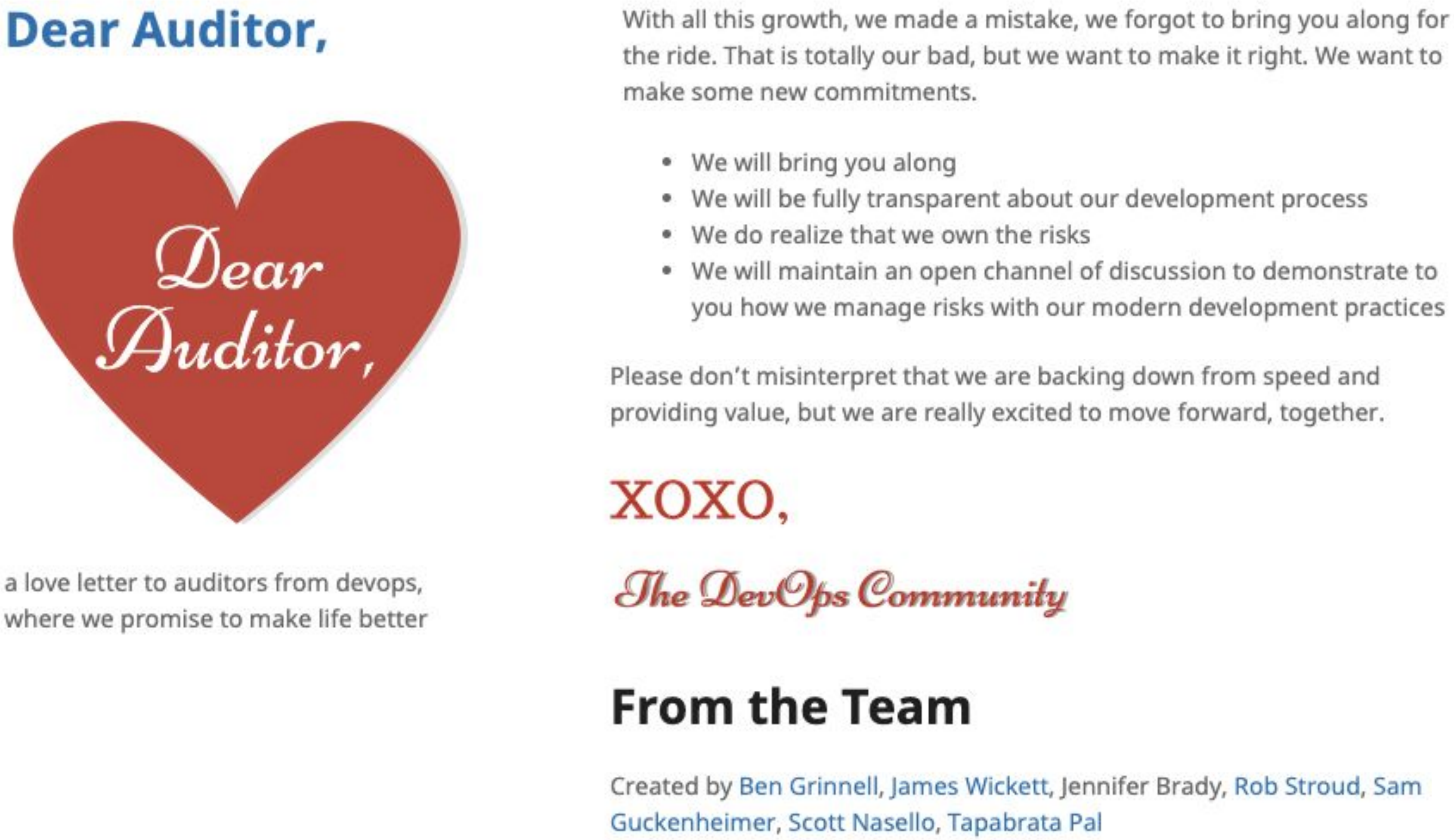
October 19, 2022

v1.0.0

Beyonce Rule

If You Like It, Then You Should Tweet On It

@BillBensing



Dear Auditor,

a love letter to auditors from devops, where we promise to make life better

With all this growth, we made a mistake, we forgot to bring you along for the ride. That is totally our bad, but we want to make it right. We want to make some new commitments.

- We will bring you along
- We will be fully transparent about our development process
- We do realize that we own the risks
- We will maintain an open channel of discussion to demonstrate to you how we manage risks with our modern development practices

Please don't misinterpret that we are backing down from speed and providing value, but we are really excited to move forward, together.

XOXO,

The DevOps Community

From the Team

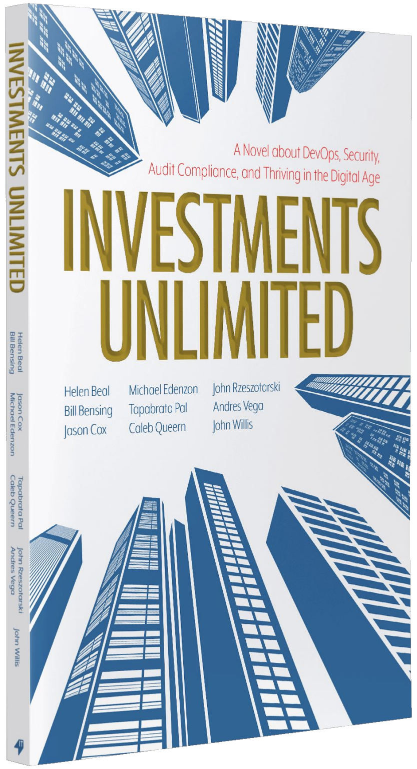
Created by Ben Grinnell, James Wickett, Jennifer Brady, Rob Stroud, Sam Guckenheimer, Scott Nasello, Tapabrata Pal

Source: https://www.dearauditor.org

> “Make the right way, the easy and default way, for anyone who wants to build software.”

Bill Bensing

Red Hat – Managing Architect – Software Factory

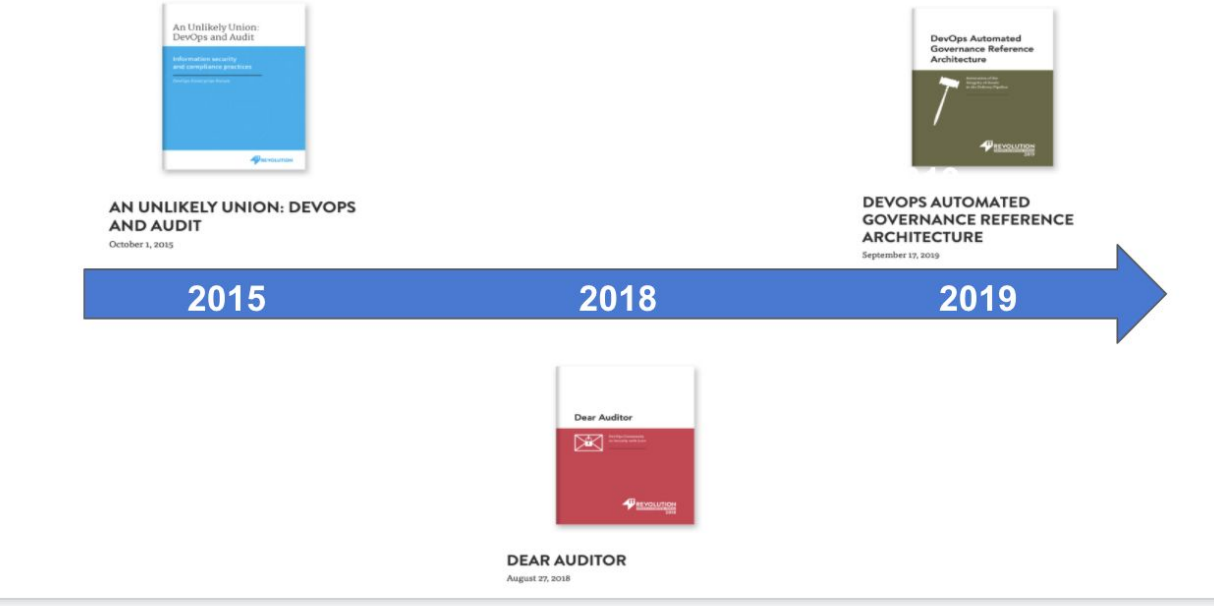


Investments Unlimited

A Novel About DevOps, Security, Audit Compliance, and Thriving in the Digital Age

By Helen Beal, Bill Bensing, Jason Cox, Michael Edenzon, Dr. Tapabrata "Topo" Pal, Caleb Queern, John Rzeszotarski, Andres Vega, and John Willis

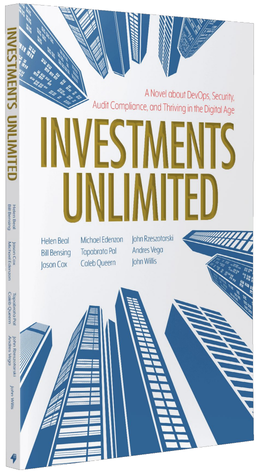
https://itrevolution.com/investments-unlimited-book


An Unlikely Union: DevOps and Audit
DevOps Automated Governance Reference Architecture
Dear Auditor
2015
2018
2019

AN UNLIKELY UNION: DEVOPS AND AUDIT
October 1, 2015

DEVOPS AUTOMATED GOVERNANCE REFERENCE ARCHITECTURE
September 17, 2019

DEAR AUDITOR
August 27, 2018



A Novel about DevOps, Security, Audit Compliance, and Thriving in the Digital Age

INVESTMENTS UNLIMITED

Helen Beal
Bill Bensing
Jason Cox
Michael Edenzon
Tapabrata Pal
Caleb Queern
John Rzeszotarski
Andres Vega
John Willis

Bottom Line Up Front

People Should Not Execute

The Governance Process

Machines Must Execute

The Governance Process

People Design, Develop, & Codify The Governance Process

Governance Refers To Security, Compliance, and Audit.

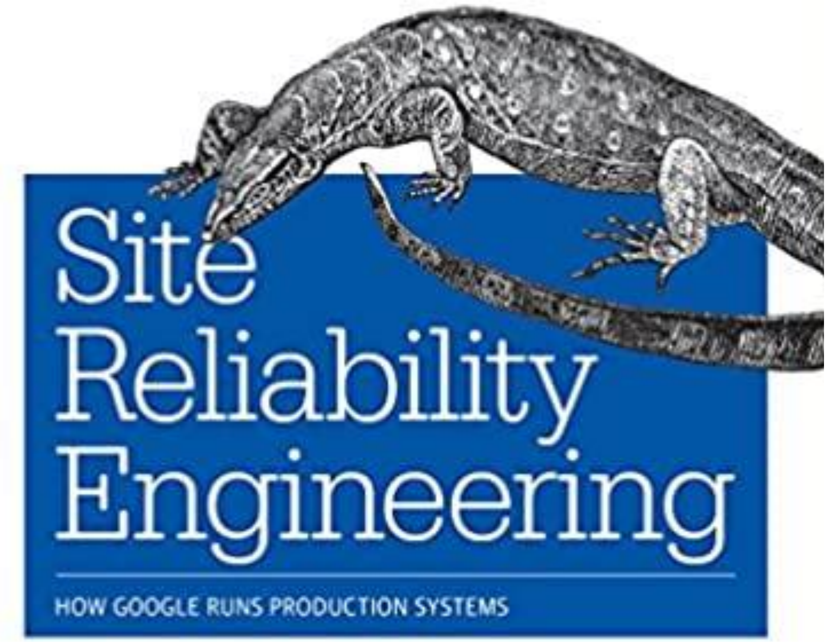
Governance Is The Current Bottleneck For Software Delivery

We Must Modernize Governance Capabilities To Address This Bottleneck

Modernizing Governance Is Automating The Governance Process

But…

It’s More Than Just Automation, It’s Autonomous


Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS

> “For SRE, automation is a force multiplier, not a panacea. Of course, just multiplying force does not naturally change the accuracy of where that force is applied: doing automation thoughtlessly can create as many problems as it solves. Therefore, while we believe that software-based automation is superior to manual operation in most circumstances, better than either option is a higher-level system design requiring neither of them—an **autonomous** system. Or to put it another way, the value of automation comes from both what it does and its judicious application.”

Site Reliability Engineer, Google

Chapter 7 - The Evolution of Automation at Google

Source: https://sre.google/sre-book/eliminating-toil/

Modern Governance Is A Higher-Level Governance System Design

Modern Governance is Autonomous Governance

Autonomous Governance Only Works With Modern Rules

“Beyond The Goal” – Dr. Eliyahu Goldratt

1

Its Power

Achieve speed-to-market & highest trust simultaneously.

2

Diminished Limitations

Ineffective manual processes which decrease time-to-market

3

Old Rules

Domain-specific people manually verify all aspects of trust: Security, Compliance, & more…

4

New Rules

Domain-specific people define & codify trust, automation validates.

Agenda

The Governance Problem

Solving the Governance Problem

A Solution – Governance As A Service

Governance & Engineering Productivity

The Governance Engineering Team

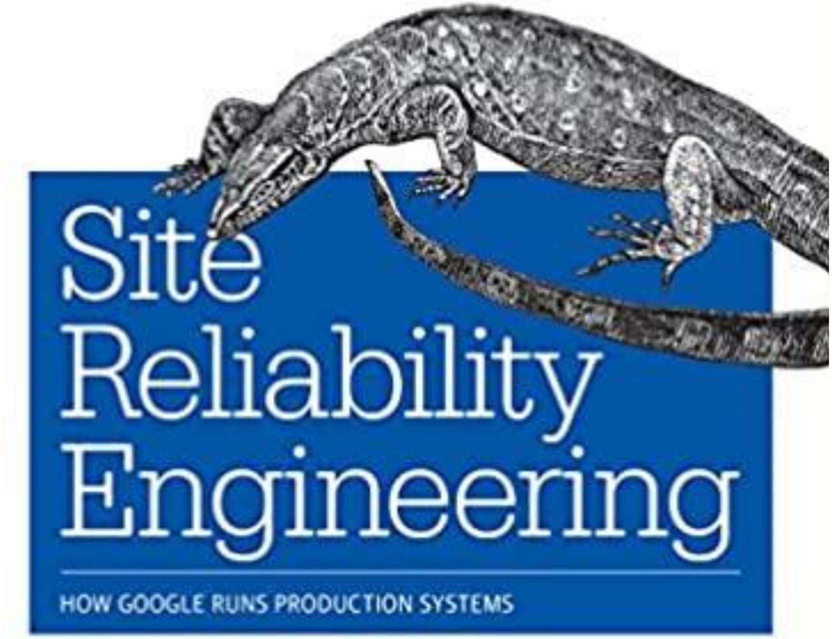
The Governance Problem

In Most Organizations, Governance is…

Security
Compliance
+ Audit

Toil

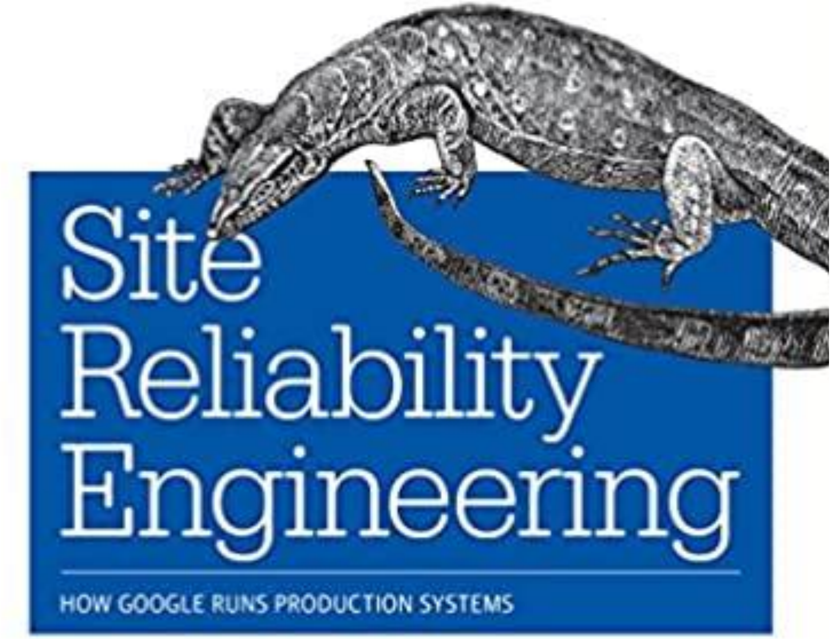
What Is This Toil?



> “Toil is the kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value, and that scales linearly as a service grows.”

Vivek Rau
Site Reliability Engineer, Google

Source: https://sre.google/sre-book/eliminating-toil/


Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS

> “If a human operator needs to touch your system during normal operations, you have a bug. The definition of normal changes as your systems grow.”

Carla Geisser
Site Reliability Engineer, Google

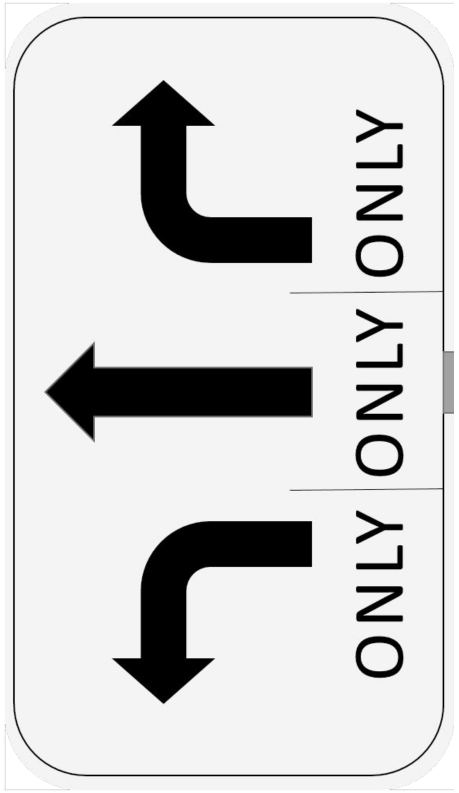
Source: https://sre.google/sre-book/eliminating-toil/

Governance Toil

Delivery Toil

Governance Toil

Humans Turning Cranks Of The Governance Process

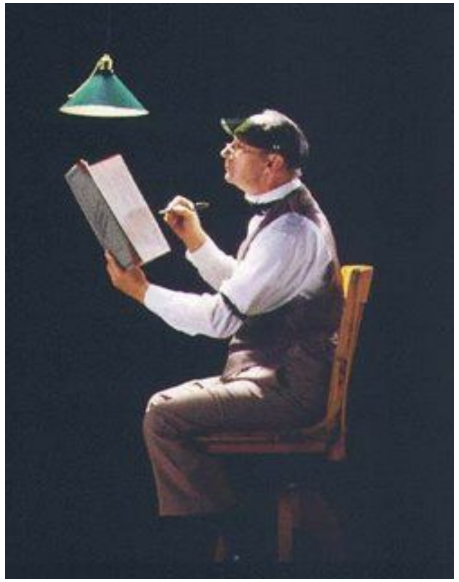


Delivery Toil

Outcomes Caused By Ambiguity of Governance Process

Because of this toil…

What is Meant To Mitigate Risks Actually Increases Risk!



I Have The Numbers To Prove It

A Very Relatable Example

Why Can’t Governance Take Just Second?!

16 hrs.
Per Change

2 Weeks

90%
Success Rate Per Change

59%
Success Rate When Changes Batched

62%
Success Rate When Changes Batched

How Do We Fix This?

Solving the Governance Problem

Automate That Stuff!

~~Automate~~ Autonomize

That Stuff!

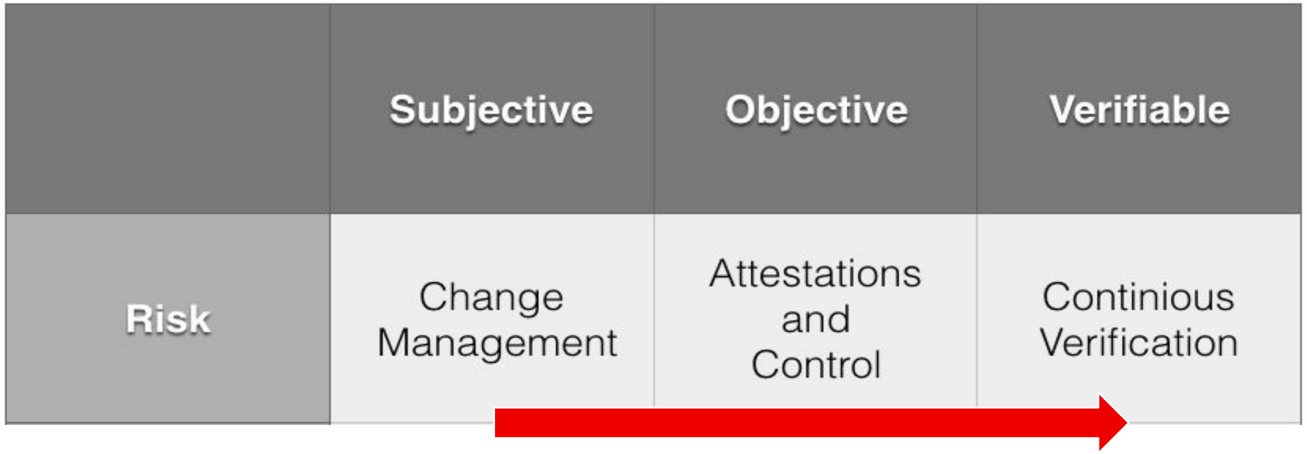
How Do We Autonomize Governance?

Five Guiding Principles

1. **Collaboration** Across All Parties: Software Engineers, Systems Operators, Security, Compliance, Auditors.
2. Develop **Enabling Constraints**
3. Require **Explicit Evidence** for an **Idempotent Process**
4. Governance Execution is **Zero-Trust**
5. Implementation Must Operate **Ephemeral** and **Immutable**

We Need To Think Differently

Autonomizing Requires Moving From Subjective to Continuous Verification



	Subjective	Objective	Verifiable
Risk	Change Management	Attestations and Control	Continious Verification

To Achieve Continuous Verification

We Must Autonomize The Human Controlled Gates

The Control Gates To Autonomize

Continuous Verification For All Go/No-Go Decision Points

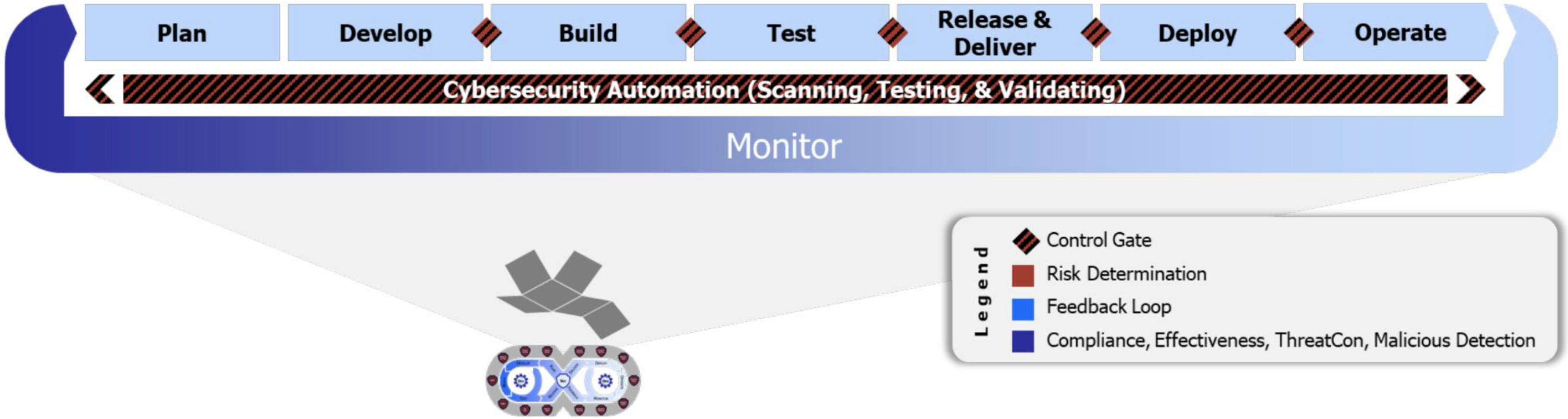
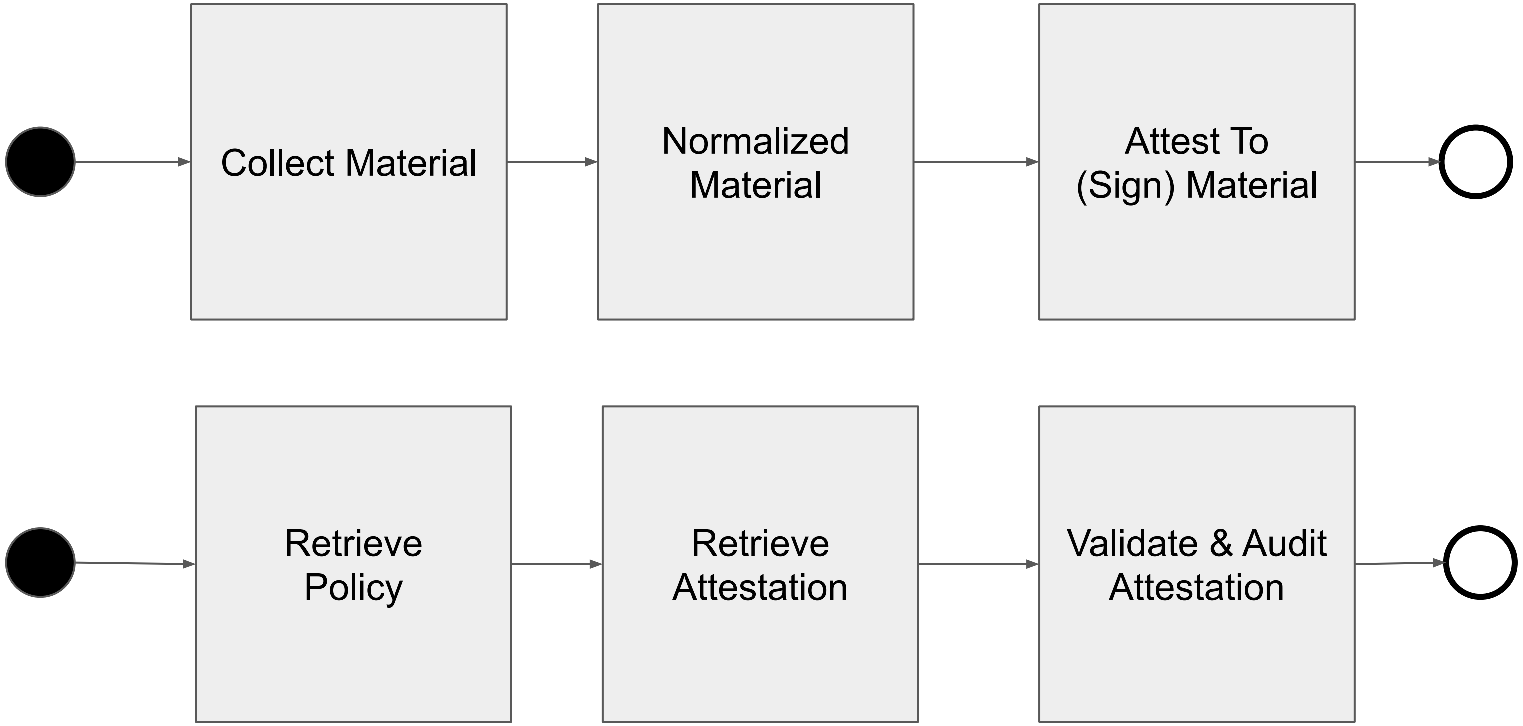

Plan
Develop
Build
Test
Release & Deliver
Deploy
Operate
Cybersecurity Automation (Scanning, Testing, & Validating)
Monitor
Legend
Control Gate
Risk Determination
Feedback Loop
Compliance, Effectiveness, ThreatCon, Malicious Detection

Figure 6 DevSecOps Lifecycle Phases, Continuous Feedback Loops, & Control Gates

Source: https://software.af.mil/wp-content/uploads/2021/05/DoD-Enterprise-DevSecOps-2.0-Strategy-Guide.pdf (Pg. 19)

How Do We Autonomize Human Control Gates?

Autonomize Control Gate Activity



Evidence & Attestation Procedure

Policy Enforcement Point Procedure

To Do This Properly,

We Need A New Concept

We Need A

Governance Contract

What Is a Governance Contract?

A Governance Contract Defines The Semantics & Syntax of Our Governance Primitives

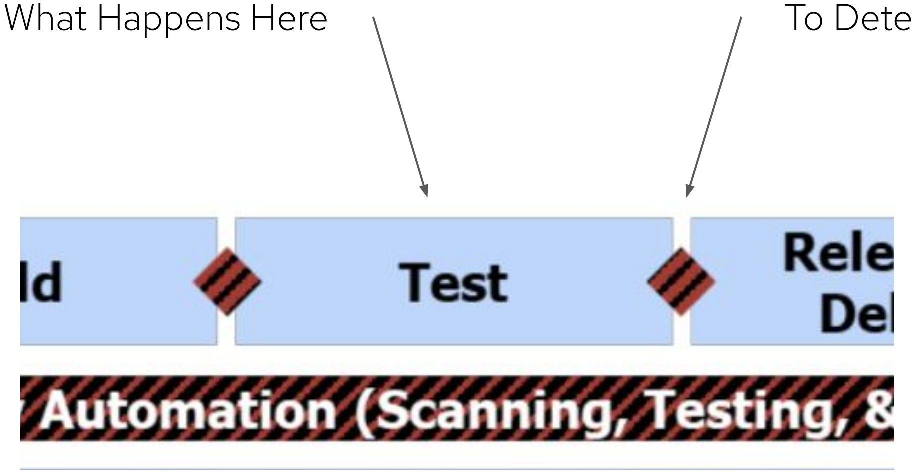
It’s How We Codify Our Governance Specifications

The Governance Contract Describes

In A Way That Is Technology & Tool Agnostic

What Happens Here

To Determine If We Pass Here



For **All Gates**, Not Just Testing

Governance Contact

The Components of a Governance Contract

```
common-vulnerability-exploits:
     attestations:
          high-severity:
               count:    1
          medium-severity:
               count:    3
          low-severity:
               count:    12
     evidence:
          material:      [...]
          signatures:    [...]
```

At A Minimum..

- The Ubiquitous Language
- Technology Agnostic
- Understandable by technical & non-technical team members
- The data exchange format for all autonomous governance implementations

Governance Contact

The Components of a Governance Contract

```
common-vulnerability-exploits:
     attestations:
          high-severity:
               count:    1
          medium-severity:
               count:    3
          low-severity:
               count:    12
     evidence:
          material:      [...]
          signatures:    [...]
```

Governance Procedure

The control gate required by the governance process.

Governance Contact

The Components of a Governance Contract

```
common-vulnerability-exploits:
     attestations:
          high-severity:
               count:    1
          medium-severity:
               count:    3
          low-severity:
               count:    12
     evidence:
          material:      [...]
          signatures:    [...]
```

Procedure Element

A specific output of the procedure which is measured for compliance to a policy.

Governance Contact

The Components of a Governance Contract

```
common-vulnerability-exploits:
     attestations:
          high-severity:
               count:    1
          medium-severity:
               count:    3
          low-severity:
               count:    12
     evidence:
          material:      [...]
          signatures:    [...]
```

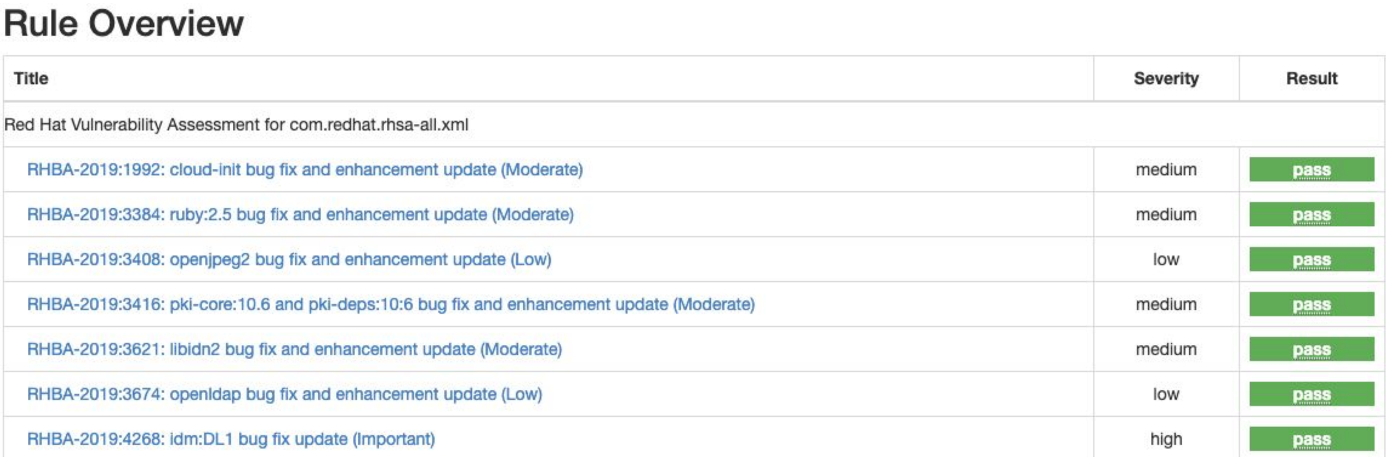
Procedure Element Value

The value which is evaluated during an audit against a policy.

How Is a Governance Contract Created?

Governance Contract is Serialized Evidence

First Step To Externalizing Policy Execution



Rule Overview

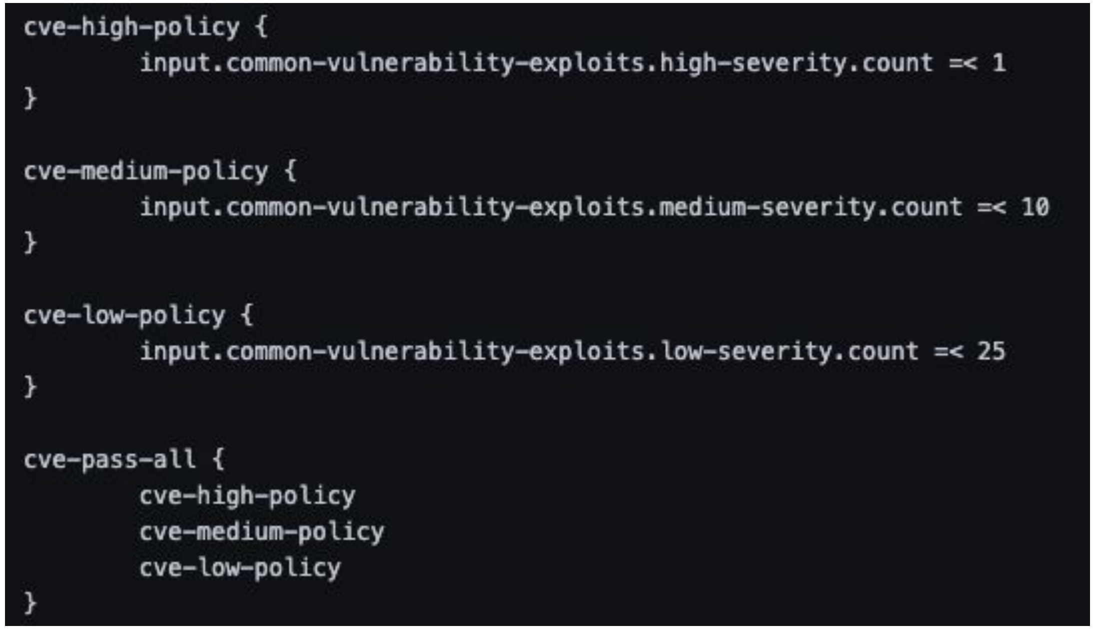
Title	Severity	Result
Red Hat Vulnerability Assessment for com.redhat.rhsa-all.xml		
RHBA-2019:1992: cloud-init bug fix and enhancement update (Moderate)	medium	pass
RHBA-2019:3384: ruby:2.5 bug fix and enhancement update (Moderate)	medium	pass
RHBA-2019:3408: openjpeg2 bug fix and enhancement update (Low)	low	pass
RHBA-2019:3416: pki-core:10.6 and pki-deps:10:6 bug fix and enhancement update (Moderate)	medium	pass
RHBA-2019:3621: libidn2 bug fix and enhancement update (Moderate)	medium	pass
RHBA-2019:3674: openldap bug fix and enhancement update (Low)	low	pass
RHBA-2019:4268: idm:DL1 bug fix update (Important)	high	pass

```
common-vulnerability-exploits:
    attestations:
        high-severity:
            count:    1
        medium-severity:
            count:    3
        low-severity:
            count:    12
    evidence:
        material:      [...]
        signatures:    [...]
```

How Is A Governance Contract Evaluated Against A Policy?

Apply Policy as Code To Governance Contract

Second Step To Externalizing Policy Execution



```
cve-high-policy {
        input.common-vulnerability-exploits.high-severity.count =< 1
}

cve-medium-policy {
        input.common-vulnerability-exploits.medium-severity.count =< 10
}

cve-low-policy {
        input.common-vulnerability-exploits.low-severity.count =< 25
}

cve-pass-all {
        cve-high-policy
        cve-medium-policy
        cve-low-policy
}
```

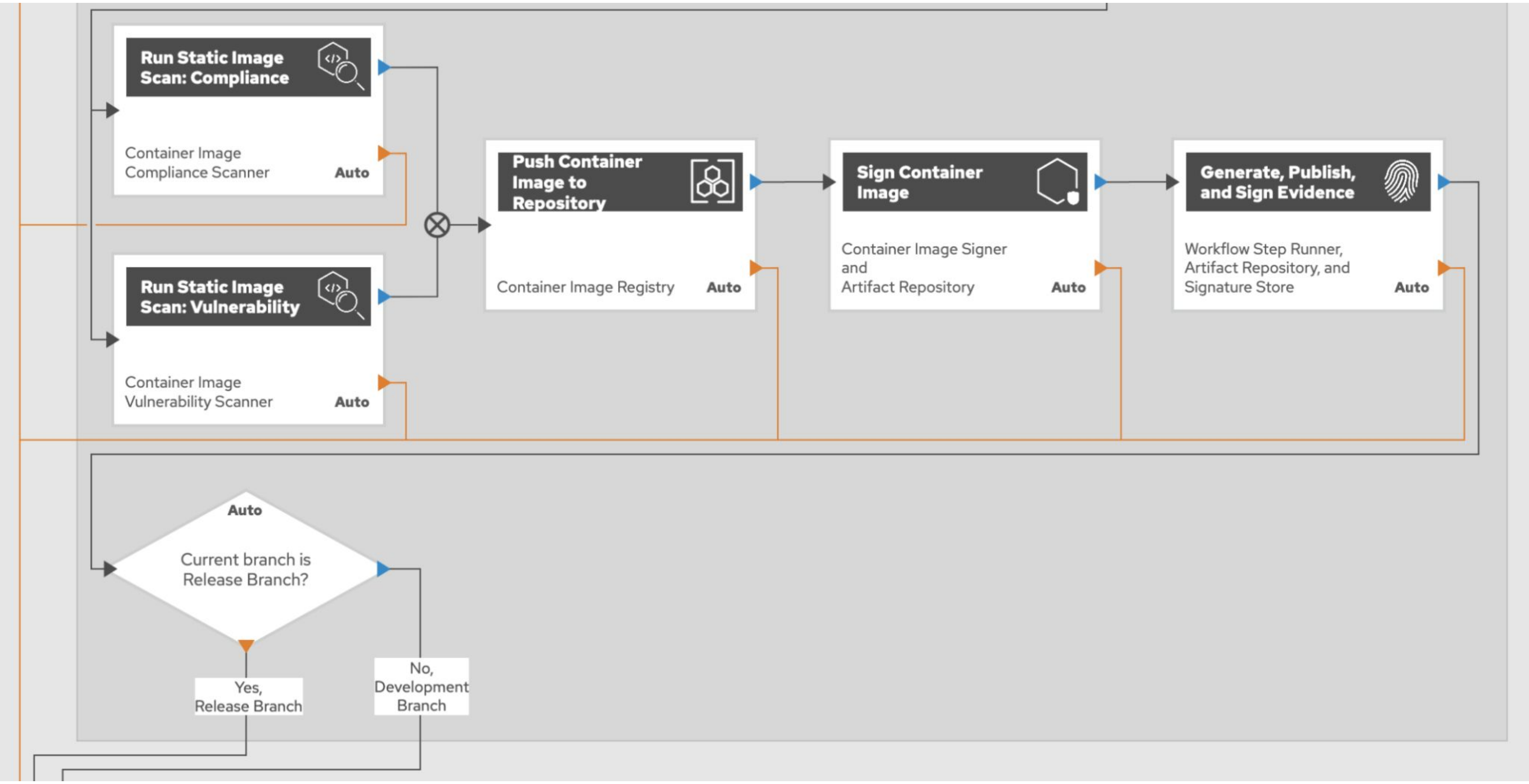
?

```
common-vulnerability-exploits:
    attestations:
        high-severity:
            count:    1
        medium-severity:
            count:    3
        low-severity:
            count:    12
    evidence:
        material:      [...]
        signatures:    [...]
```

What Does This Look Like When Applied to Software Delivery?

Continuous Integration as Evidence

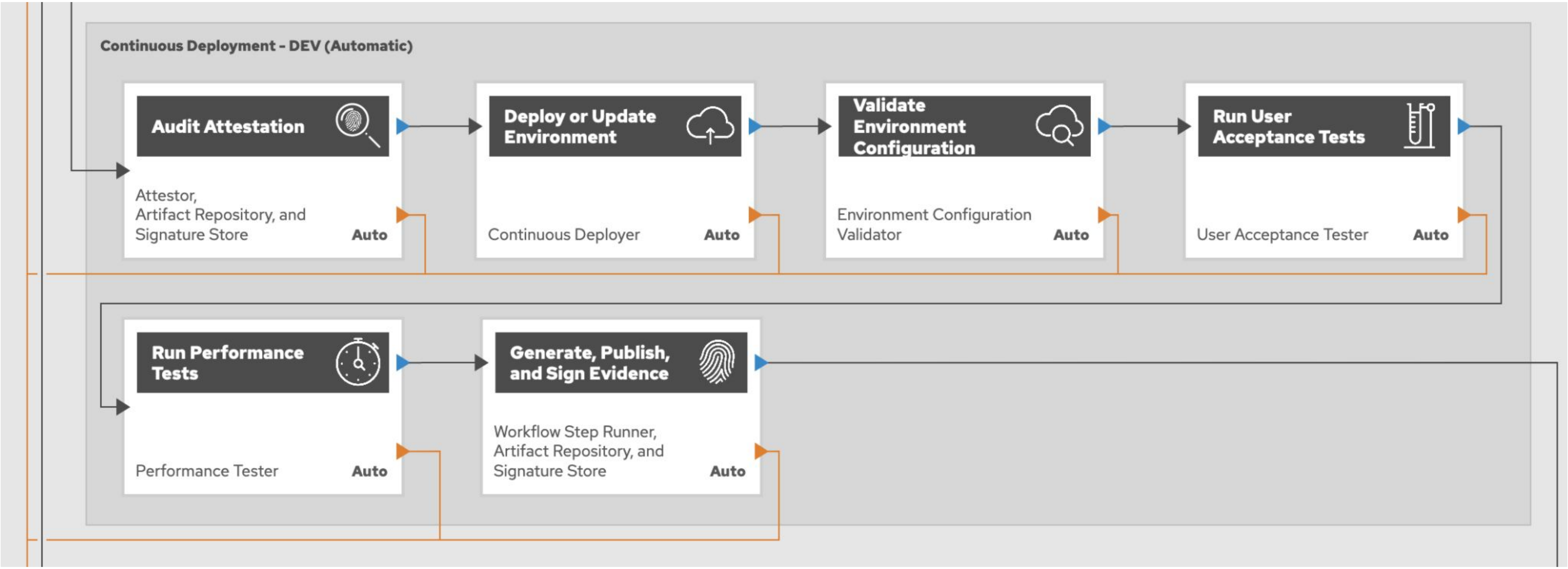
Collection & Attestation of Continuous Integration



Run Static Image Scan: Compliance
Container Image Compliance Scanner
Auto
Run Static Image Scan: Vulnerability
Container Image Vulnerability Scanner
Auto
Push Container Image to Repository
Container Image Registry
Auto
Sign Container Image
Container Image Signer and Artifact Repository
Auto
Generate, Publish, and Sign Evidence
Workflow Step Runner, Artifact Repository, and Signature Store
Auto
Auto
Current branch is Release Branch?
Yes, Release Branch
No, Development Branch

Validateable Continuous Deployment

Audits Are Autonomous Pre-Conditions of Continuous Deployment



Continuous Deployment - DEV (Automatic)

Audit Attestation
Attestor, Artifact Repository, and Signature Store — **Auto**

Deploy or Update Environment
Continuous Deployer — **Auto**

Validate Environment Configuration
Environment Configuration Validator — **Auto**

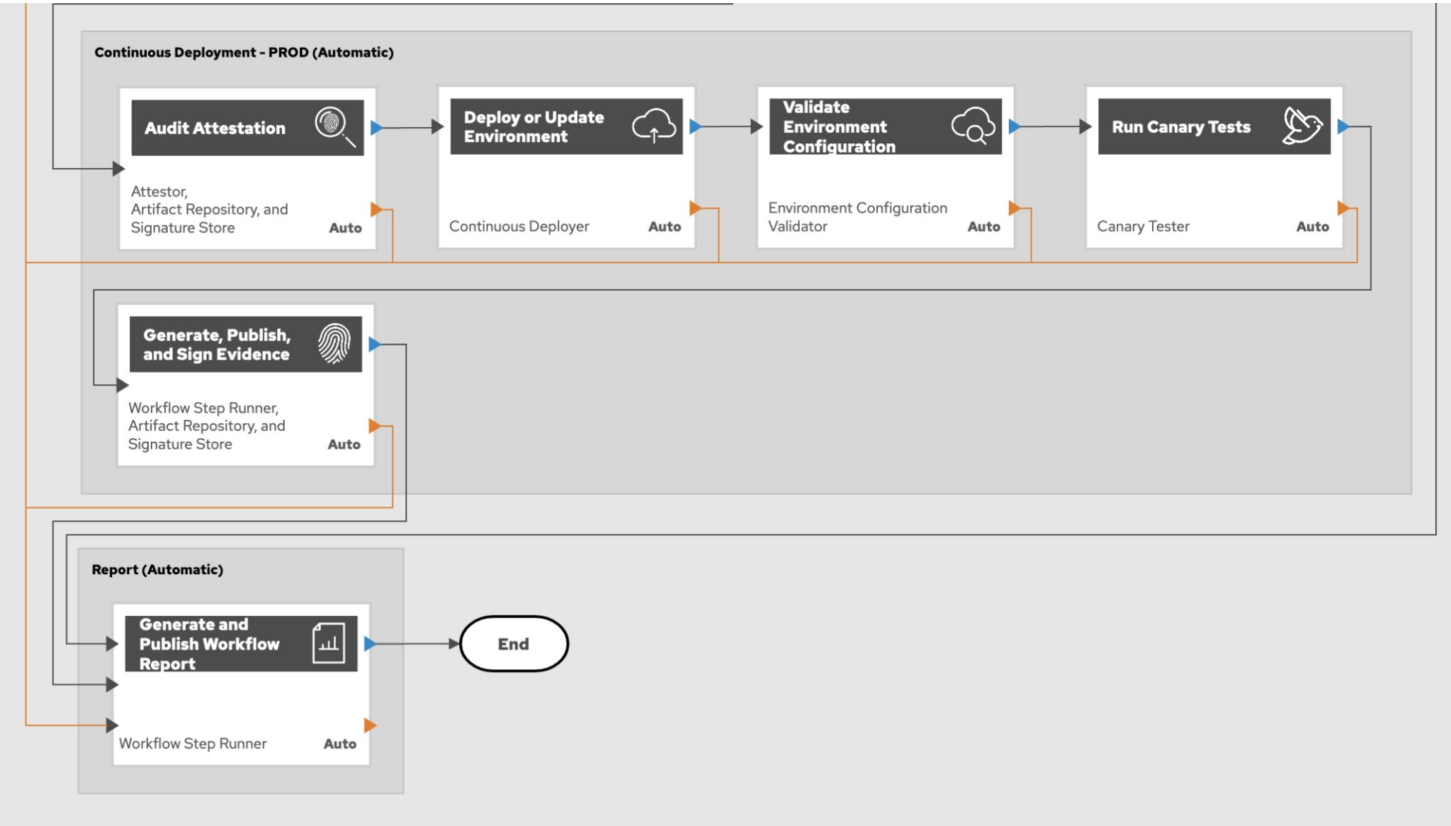
Run User Acceptance Tests
User Acceptance Tester — **Auto**

Run Performance Tests
Performance Tester — **Auto**

Generate, Publish, and Sign Evidence
Workflow Step Runner, Artifact Repository, and Signature Store — **Auto**

100% Autonomous - Commit to Production

Autonomous Governance = Compliance as Code + Policy as Code


Continuous Deployment - PROD (Automatic)
Audit Attestation
Attestor, Artifact Repository, and Signature Store
Auto
Deploy or Update Environment
Continuous Deployer
Auto
Validate Environment Configuration
Environment Configuration Validator
Auto
Run Canary Tests
Canary Tester
Auto
Generate, Publish, and Sign Evidence
Workflow Step Runner, Artifact Repository, and Signature Store
Auto
Report (Automatic)
Generate and Publish Workflow Report
Workflow Step Runner
Auto
End

Governance as a Service - The Business Outcome

SOC2 & SOC3 - Continuous Verification of Type 1 & Type 2

5 Trust Services Criteria

1. Security
2. Availability
3. Processing Integrity
4. Confidentiality
5. Privacy

https://us.aicpa.org/content/dam/aicpa/interestareas/frc/assuranceadvisoryservices/downloadabledocuments/trust-services-criteria.pdf

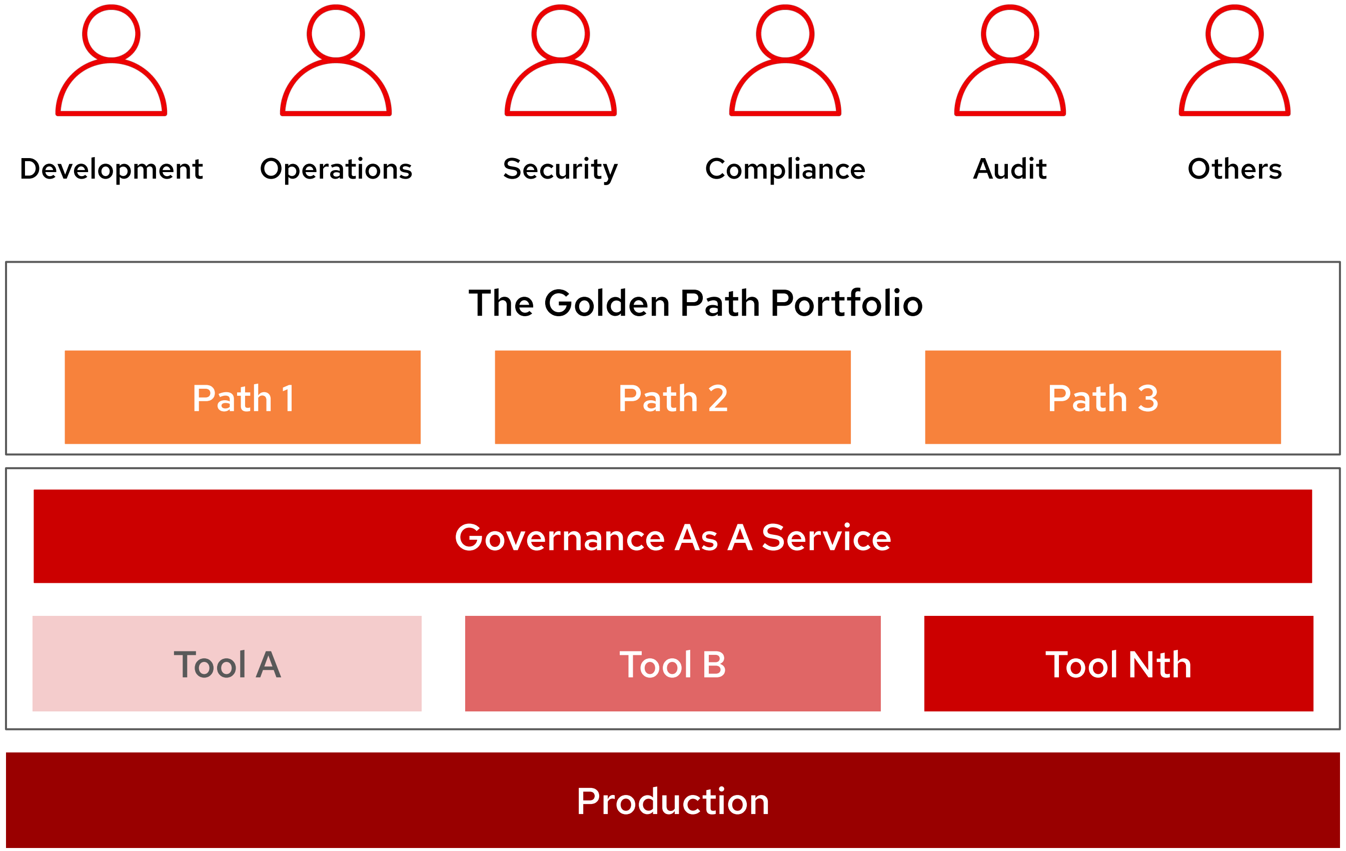
Autonomize Gates

Includes, But Not Limited Too

- Code Review Validation
- Unit Testing
- Static Code Analysis
- Dynamic Code Analysis
- Vulnerability Testing
- Compliance Validation
- Software Bill of Material (SBOM)
- Security Technical Implementation Guide (STIG)
- Use Acceptance Testing

A Solution
Governance As A Service

Golden Paths



Development

Operations

Security

Compliance

Audit

Others

The Golden Path Portfolio

Path 1

Path 2

Path 3

Governance As A Service

Tool A

Tool B

Tool Nth

Production

Golden Paths

Solve software delivery with a software engineering approach.

Creating Golden Paths which are paved on-roads for an organization.

Truly mitigate risk and reduce total cost of ownership.

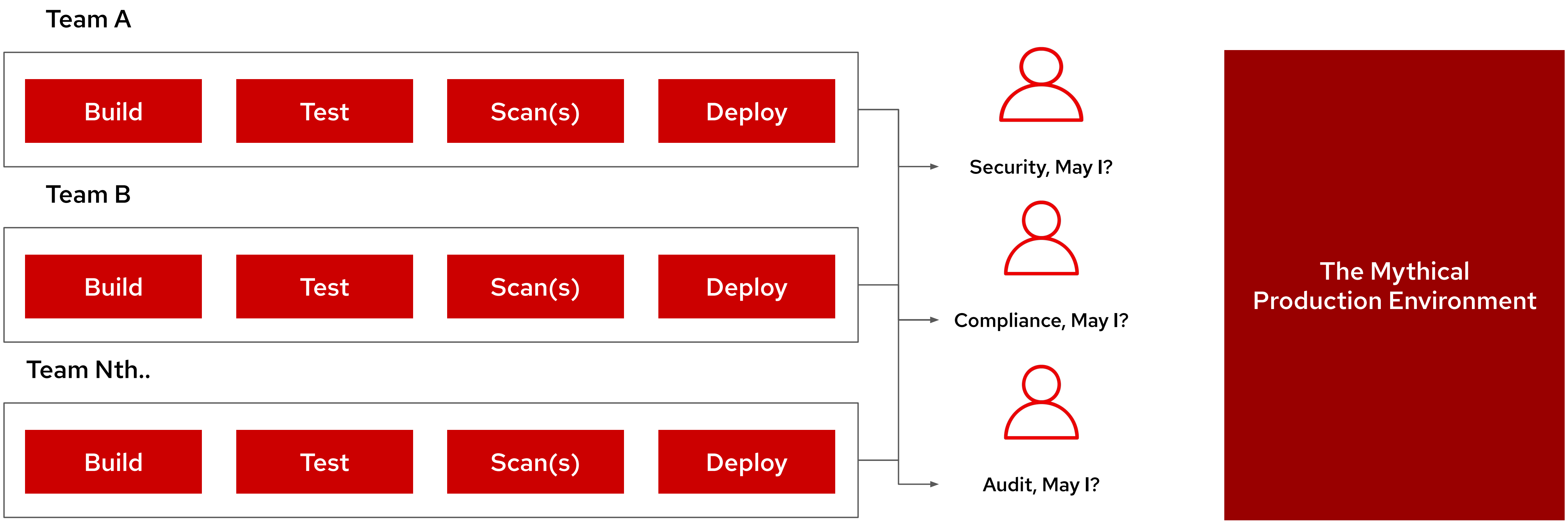
Golden Paths != Golden Cages

Let’s Build a

Golden Path to Production.

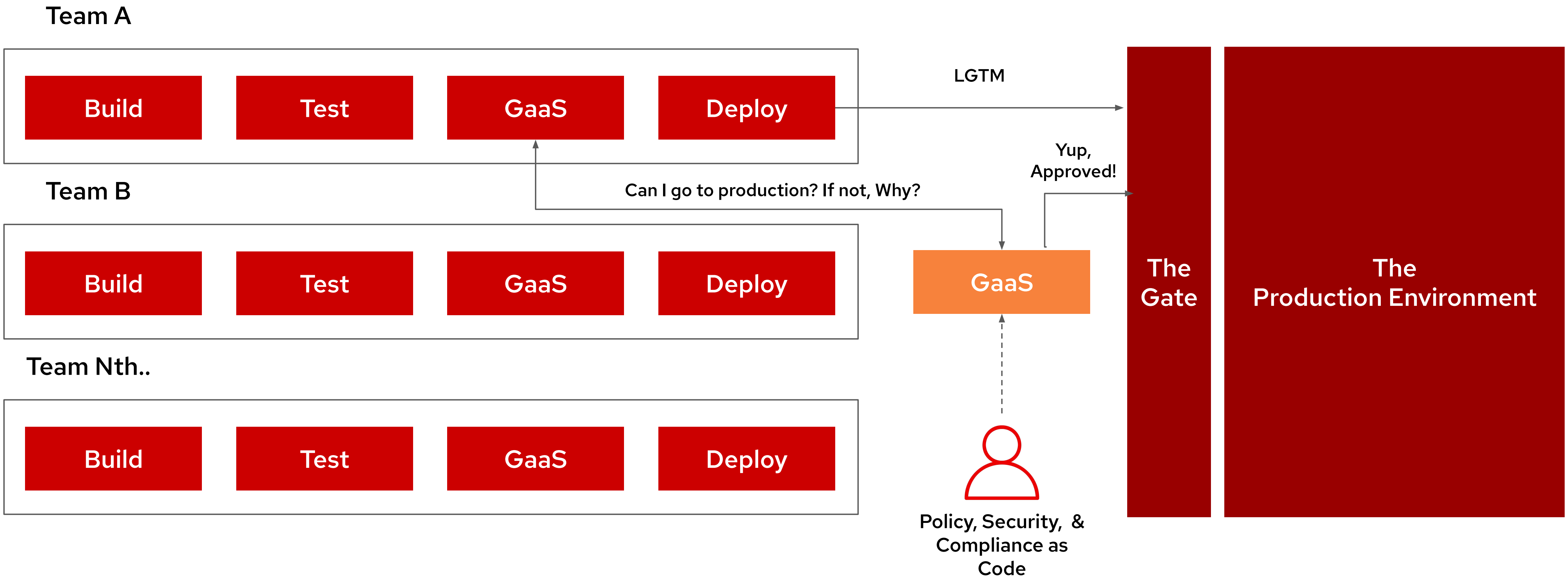
Current State

The “Free For All, File A Ticket” Approach



Future State

The Platform Approach



Zero Trust Applied To SDLC Governance

NIST SP 800-207 Zero Trust Architecture

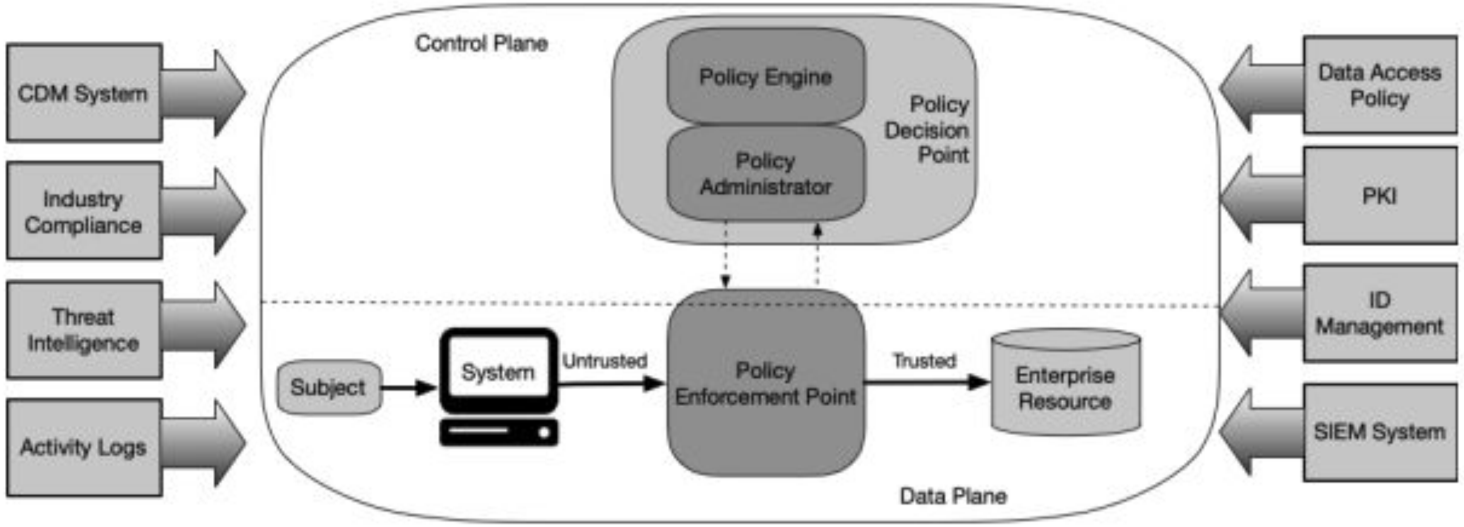
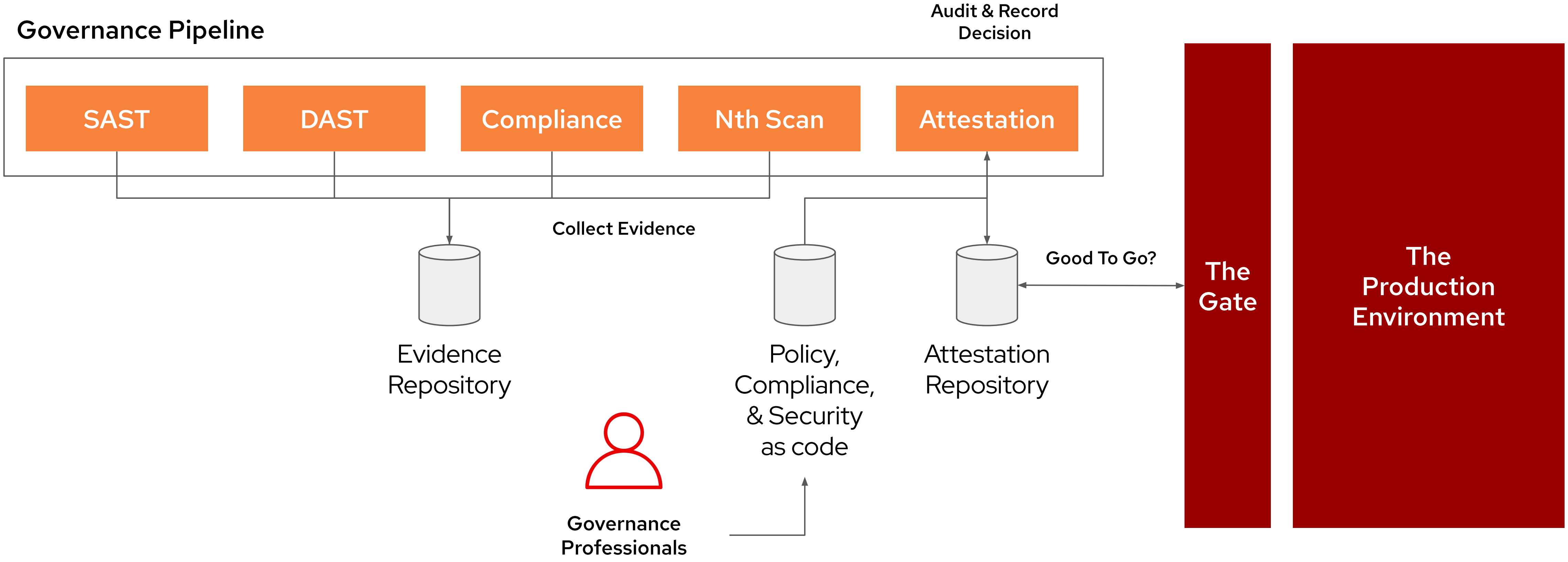


Figure 2: Core Zero Trust Logical Components

NIST SP 800-207 - Pg. 18

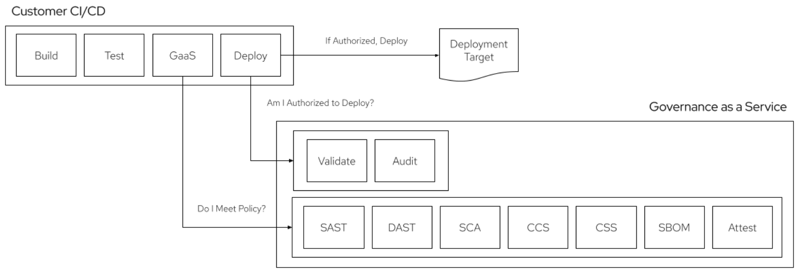
Governance As A Service

The Platform Approach



Governance As A Service

A Very Detailed Example



- Static Application Security Testing (SAST),
- Dynamic Application Security Testing (DAST),
- Software Composition Analysis (SCA),
- Container Compliance Scanning (CCS),
- Container Security Scanning (CSS),
- Software Bill of Material (SBOM) generation,
- Audit Attestation

Making The Right Thing the Easy (Default) Thing

SOC2 & SOC3 - Continuous Verification of Type 1 & Type 2

5 Trust Services Criteria

1. Security
2. Availability
3. Processing Integrity
4. Confidentiality
5. Privacy

https://us.aicpa.org/content/dam/aicpa/interestareas/frc/assuranceadvisoryservices/downloadabledocuments/trust-services-criteria.pdf

Governance & Engineering Productivity

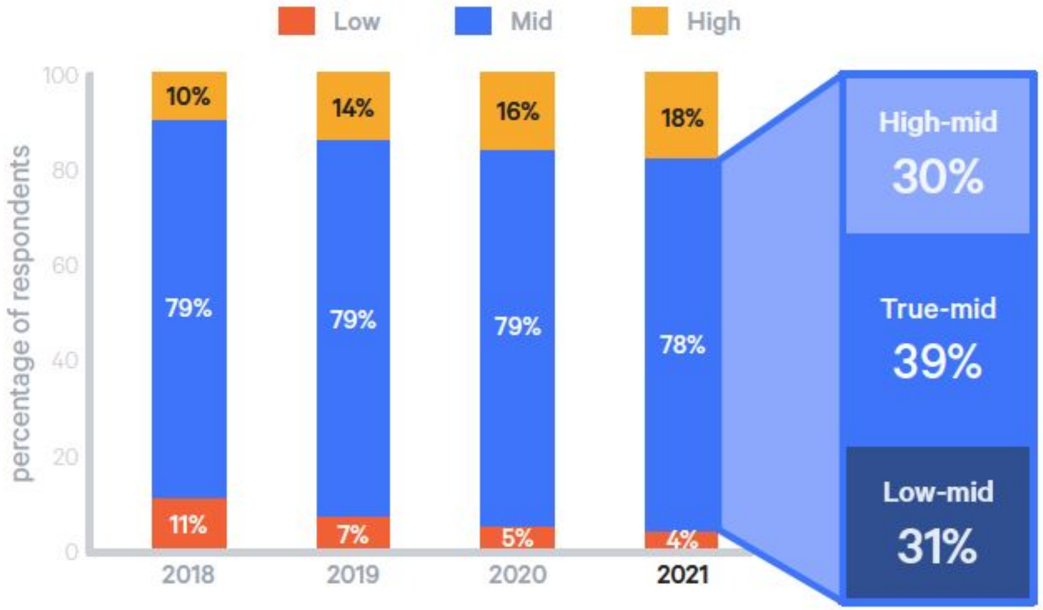
Problem - The Industry Is Stalling, It's Stuck In the Middle

Focused on Mechanics, Although Did Not Focus On Change

What Is The Middle?

- They Have
  - Laid a DevOps foundation
  - Introduced automated testing, version control, and CI/CD
  - Hired or retained teams for new approach
- Why Stuck?
  - Have not created cultures of knowledge

The vast majority remain stuck in mid-level DevOps evolution

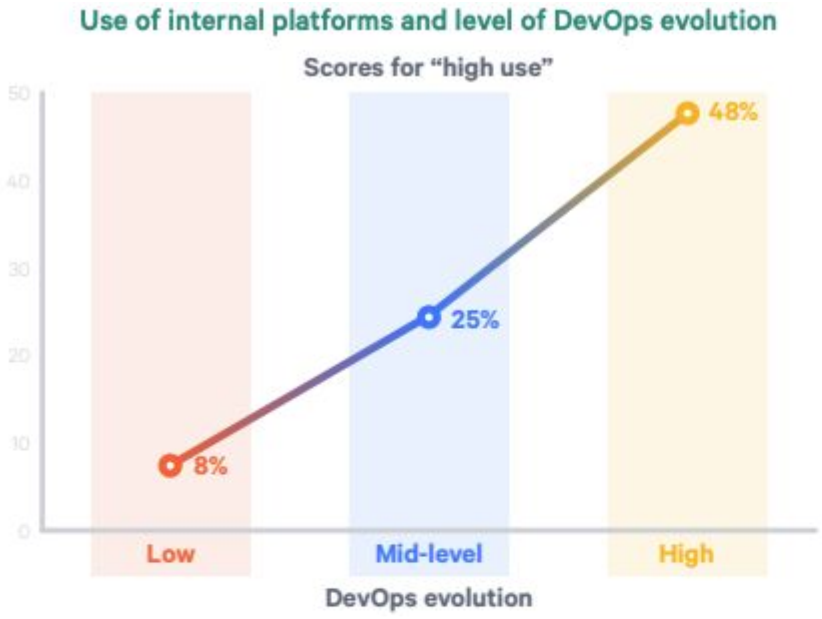


The Commonality – Internal Development Platforms

How The "Good Ones" Operate

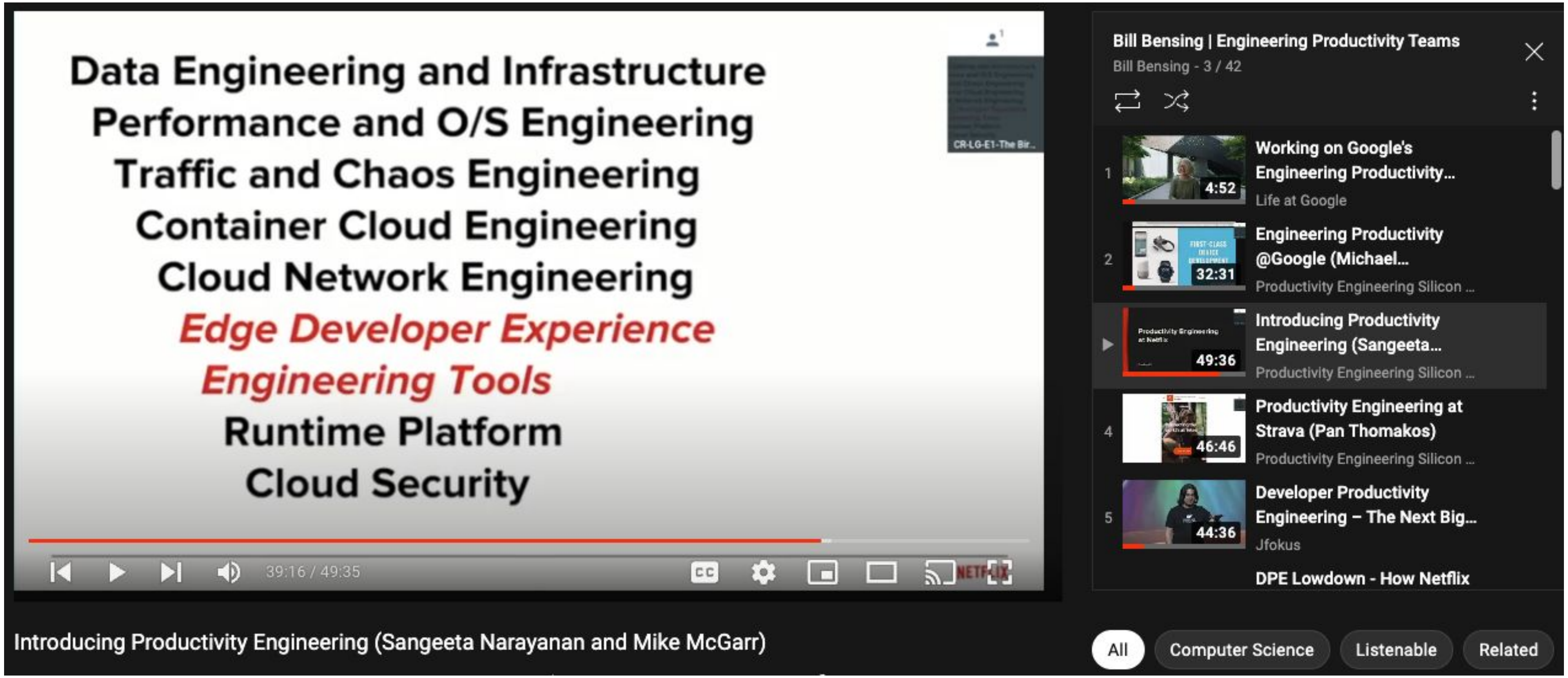
Platform Team Model

"In particular we've seen the vast majority of these organizations have adopted the platform team model that we first covered in the 2020 State of DevOps Report, where we found a high degree of correlation between DevOps evolution and the use of internal platforms." (Pg. 15)



Engineering Productivity

How the FAANGs Keep Winning



Data Engineering and Infrastructure
Performance and O/S Engineering
Traffic and Chaos Engineering
Container Cloud Engineering
Cloud Network Engineering
Edge Developer Experience
Engineering Tools
Runtime Platform
Cloud Security

39:16 / 49:35

Introducing Productivity Engineering (Sangeeta Narayanan and Mike McGarr)

Bill Bensing | Engineering Productivity Teams
Bill Bensing - 3 / 42

1 Working on Google's Engineering Productivity... 4:52
Life at Google

2 Engineering Productivity @Google (Michael... 32:31
Productivity Engineering Silicon ...

▶ Introducing Productivity Engineering (Sangeeta... 49:36
Productivity Engineering Silicon ...

4 Productivity Engineering at Strava (Pan Thomakos) 46:46
Productivity Engineering Silicon ...

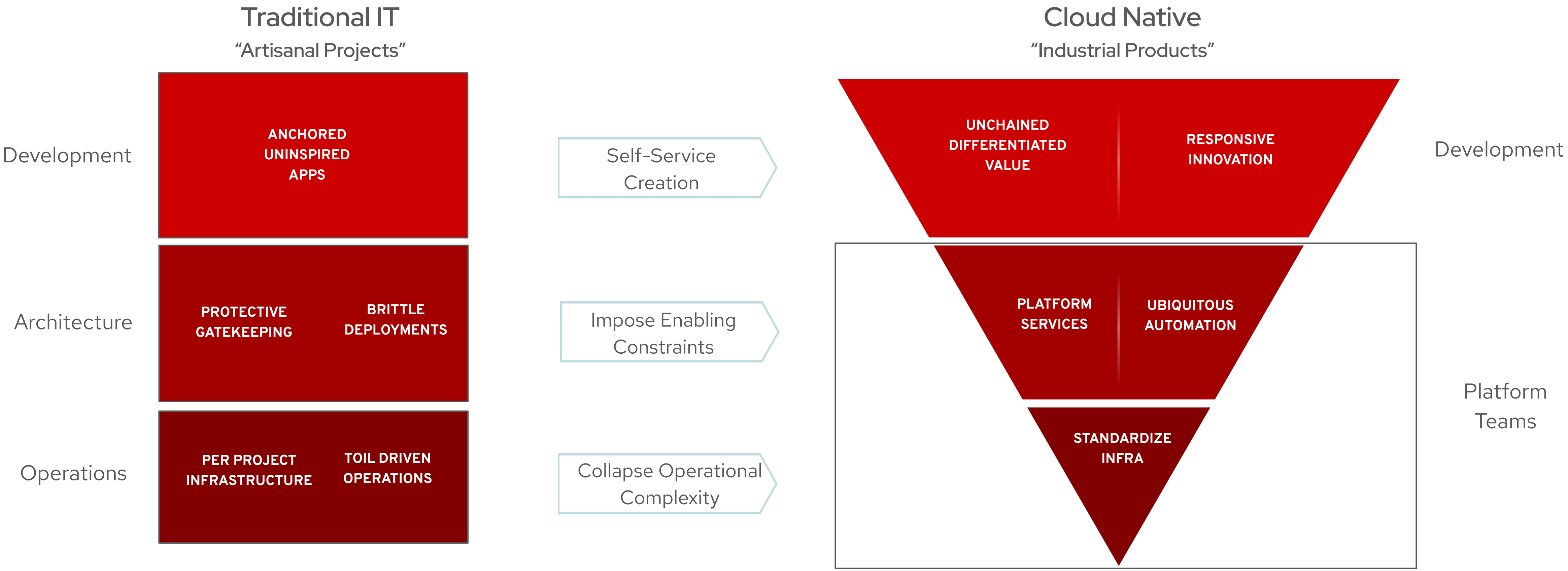
5 Developer Productivity Engineering – The Next Big... 44:36
Jfokus

DPE Lowdown - How Netflix

https://www.youtube.com/playlist?list=PLc2vHWAyCS9RB6r6sCf0AB1YlQP1q_Gma
https://www.youtube.com/watch?v=u4MADwJbxpw

The Modern Technology Organization

A Funded Focus on Engineering Productivity



Traditional IT

"Artisanal Projects"

Cloud Native

"Industrial Products"

Development

ANCHORED UNINSPIRED APPS

Self-Service Creation

UNCHAINED DIFFERENTIATED VALUE

RESPONSIVE INNOVATION

Development

Architecture

PROTECTIVE GATEKEEPING

BRITTLE DEPLOYMENTS

Impose Enabling Constraints

PLATFORM SERVICES

UBIQUITOUS AUTOMATION

Platform Teams

Operations

PER PROJECT INFRASTRUCTURE

TOIL DRIVEN OPERATIONS

Collapse Operational Complexity

STANDARDIZE INFRA

An Engineering Productivity Focus

How To Get From Old Rules to New Rules

1

Its Power

Achieve speed-to-market & highest trust simultaneously.

2

Diminished Limitations

Ineffective manual processes which decrease time-to-market

3 → 4

Old Rules

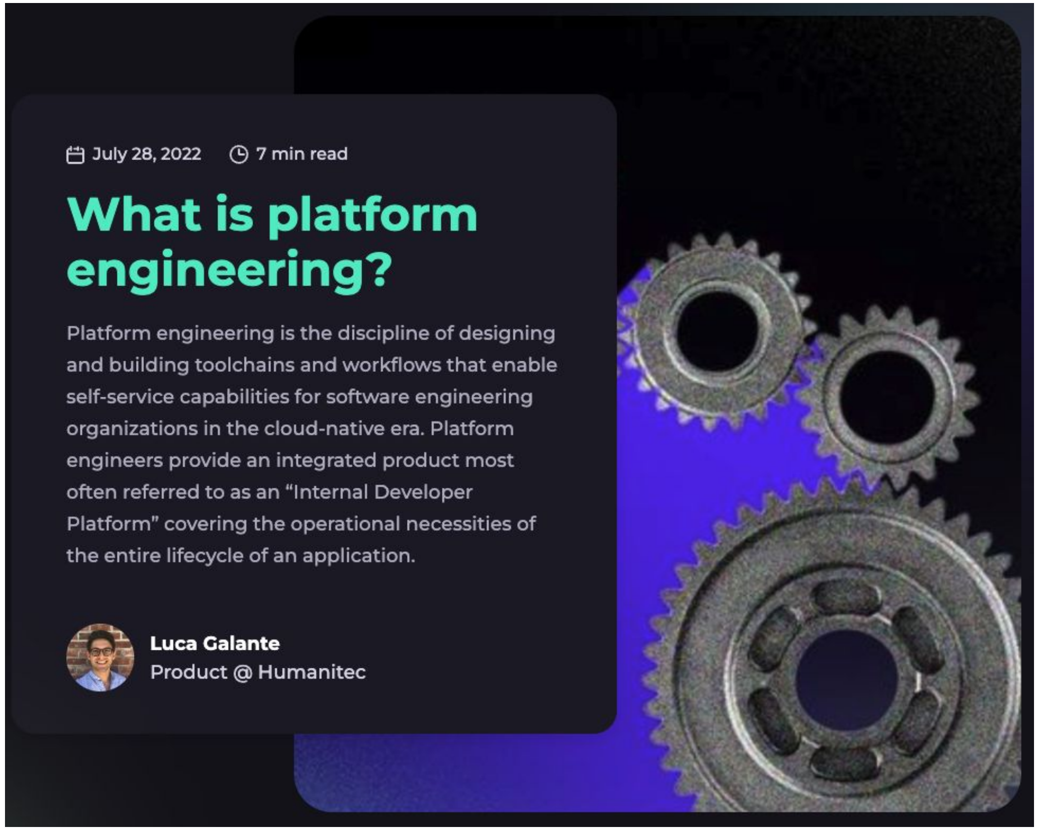
Domain-specific people manually verify all aspects of trust: Security, Compliance, & more...

New Rules

Domain-specific people define & codify trust, automation validates.

The Deliverables of an Engineering Productivity Strategy

Tactical Results For Enabling Constraints With Golden Paths

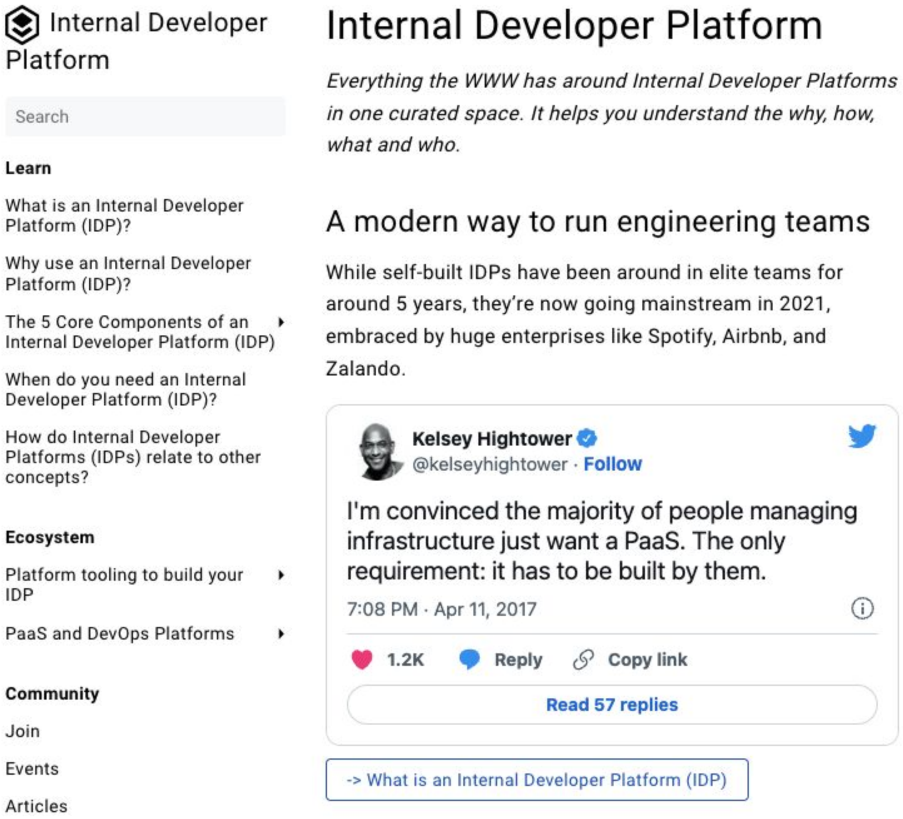


July 28, 2022 · 7 min read

What is platform engineering?

Platform engineering is the discipline of designing and building toolchains and workflows that enable self-service capabilities for software engineering organizations in the cloud-native era. Platform engineers provide an integrated product most often referred to as an “Internal Developer Platform” covering the operational necessities of the entire lifecycle of an application.

Luca Galante
Product @ Humanitec



Learn

- What is an Internal Developer Platform (IDP)?
- Why use an Internal Developer Platform (IDP)?
- The 5 Core Components of an Internal Developer Platform (IDP)
- When do you need an Internal Developer Platform (IDP)?
- How do Internal Developer Platforms (IDPs) relate to other concepts?

Ecosystem

- Platform tooling to build your IDP
- PaaS and DevOps Platforms

Community

- Join
- Events
- Articles

Internal Developer Platform

Everything the WWW has around Internal Developer Platforms in one curated space. It helps you understand the why, how, what and who.

A modern way to run engineering teams

While self-built IDPs have been around in elite teams for around 5 years, they’re now going mainstream in 2021, embraced by huge enterprises like Spotify, Airbnb, and Zalando.

Kelsey Hightower @kelseyhightower · Follow

I'm convinced the majority of people managing infrastructure just want a PaaS. The only requirement: it has to be built by them.

7:08 PM · Apr 11, 2017

-> What is an Internal Developer Platform (IDP)

https://platformengineering.org/
https://internaldeveloperplatform.org/

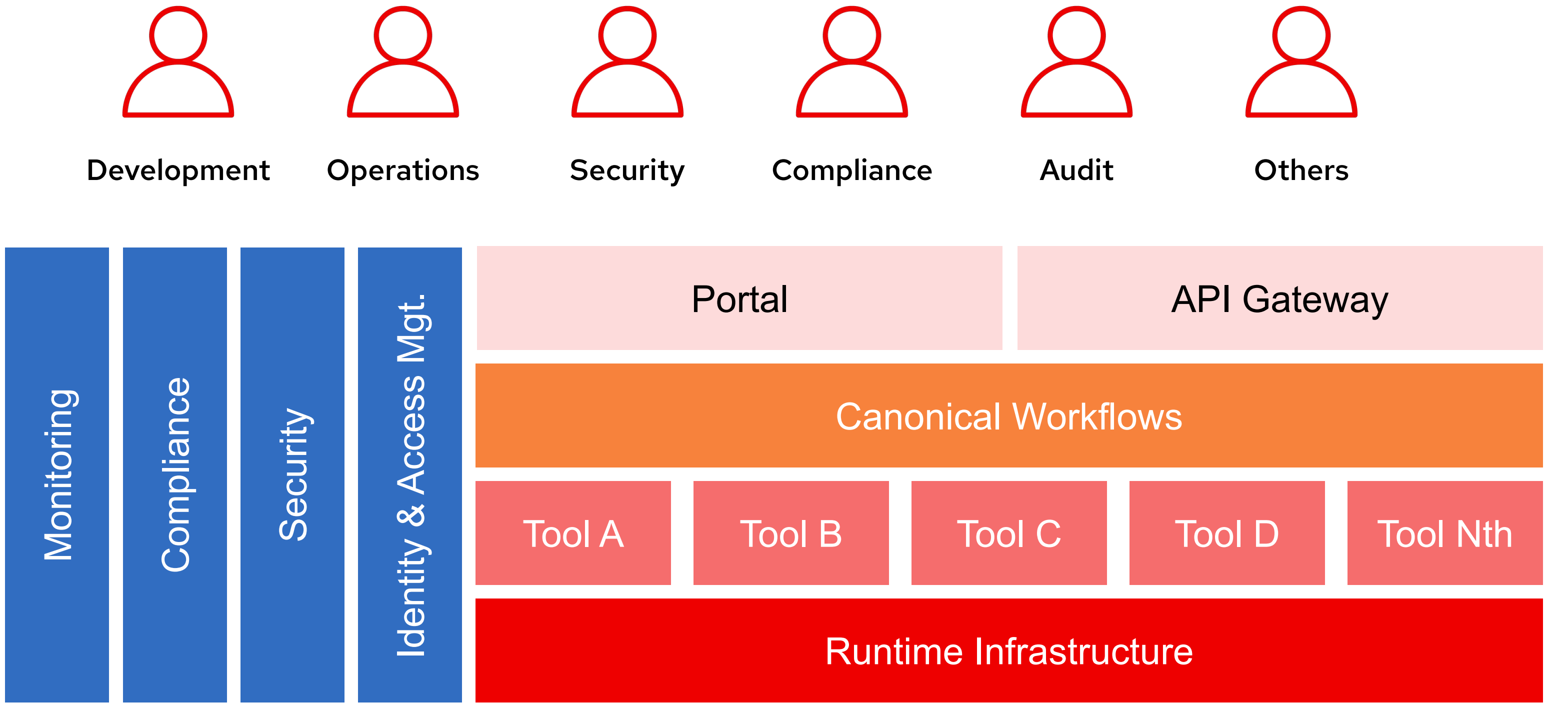
Know What Business Problem Engineering Productivity Solves?

How To Grow Top-Line Outcomes For a Company

Make The Right Thing The Easy (Default) Thing To Do

IDP For Highly-Regulated Software Delivery

Reference Architecture



Canonical Workflows - The Golden Paths

Day 1 & Day 2

Day 1 - On & Off Boarding

- New Engineer (or Other) To Organization
- New Application
- Existing Application Migration
- Off Boarding Engineering

Day 2 - CI, CD, & Operations

- Continuous Integration - LOB, Edge/Iot, & ML
- Continuous Deployment - LOB, Edge/Iot, & ML
- Governance As A Service

The Modern Governance Engineering Team

Adopt The Mindset Of Engineering Productivity

Repurpose Your Change Approval Board

Replace Your CAB With A

Governance Engineering Team

Site Reliability Engineering Principals =

Governance Engineering Principals

Measuring Modern Governance - 4 Golden Signals

SRE Golden Signals Applied to Autonomous Governance

1. **Human Touch Points** - Qty. of touch human interactions between commit and production deployment.
2. **Audit Takt Time** - Time between the start of an audit and completion of the audit; does not include remediation time.
3. **Control Ambiguity** - The quantity of governance controls which you cannot tell if they are, or are not, applicable.
4. **Control Coverage** - The quantity of applicable governance controls that are automated.

v1.0.0

The Four Golden Signals

Modern Governance Hierarchy

SRE Golden Signals Applied to Autonomous Governance

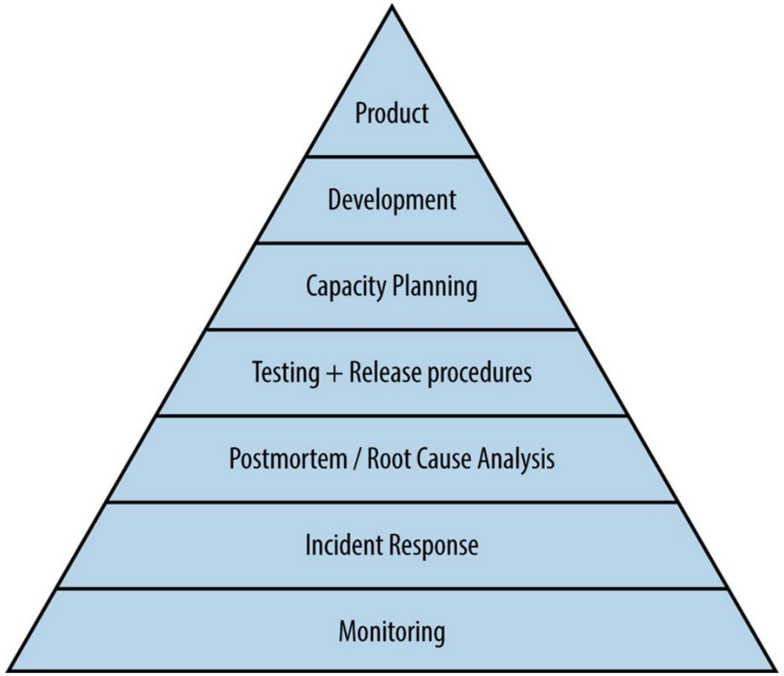
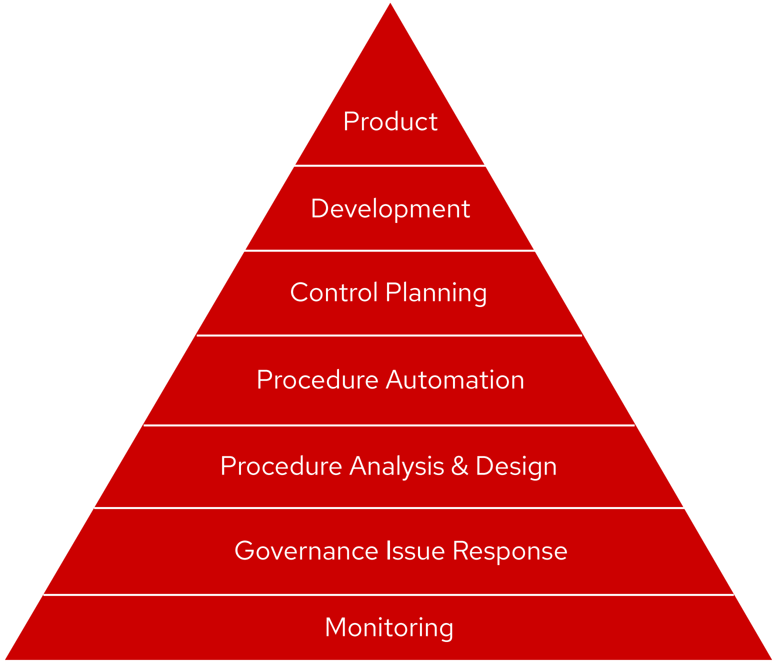


Figure III-1. Service Reliability Hierarchy



SRE Book, Part III, Practices

Governance Level Indicators

Applying SLI, SLO, and SLAs to Governance

1. Governance Level Indicator
2. Governance Level Objective
3. Governance Level Agreement

Governance & Golden Paths as Internal Products

The Governance Engineering Team

Golden Paths

- Automate Governance
- Investment To Automate Occurs Upfront
- Canonical Implementations (80/20)

Exception Paths

- Manual Evaluation
- Costs Incurred For Each CAB session
- Appropriate For Some Situations

Not Matter Road Traveled

Apply The Same Governance

Modernize Your Governance With Autonomous Governance

Autonomize Your Governance With A Governance Engineering Team

No Questions
Just Conversations

Bill Bensing

Home	billbensing.com
LinkedIn	linkedin.com/in/billbensing
Twitter	@BillBensing

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat