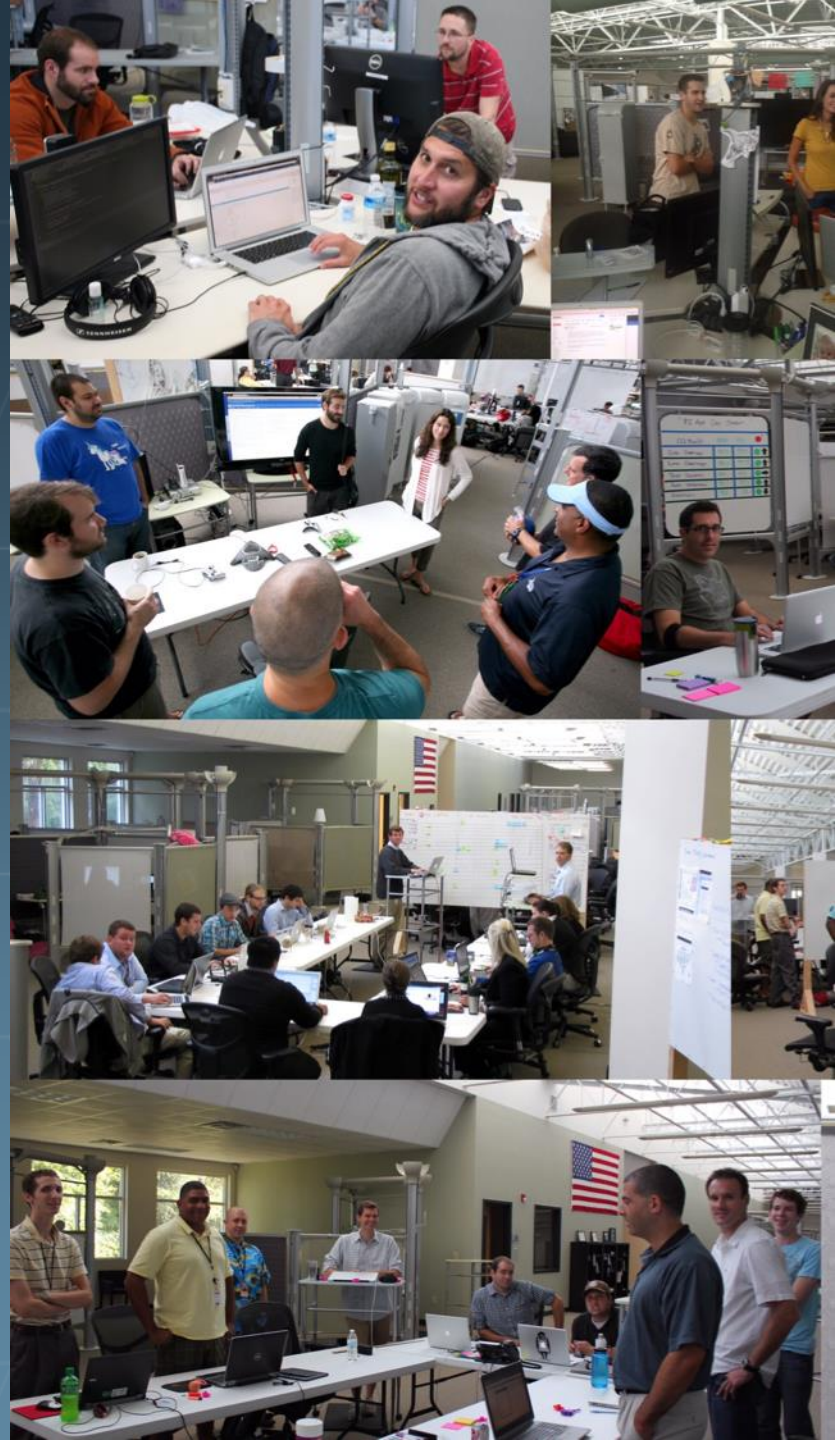
SOLUTIONS DELIVERY PLATFORM

LEVERAGING OPENSHIFT TO ACCELERATE ATO'S

Josh Boyd | Steven Terrana

Red Hat Summit 2018



AGENDA

- ❖ GETTING AN ATO IS HARD
- ❖ HOW OPENSHIFT ACCELERATES
- ❖ THE TRUSTED SUPPLY CHAIN
- ❖ SOLUTIONS DELIVERY PLATFORM
- ❖ CONTINUOUS DELIVERY AT SCALE
- ❖ DEMO

WHAT IS AN ATO?

Getting an Authority to Operate requires documenting over 1,500 security controls aggregated from multiple sources.

FISMA
The Federal Information Security Management Act

NIST
The National Institute of Standards and Technology

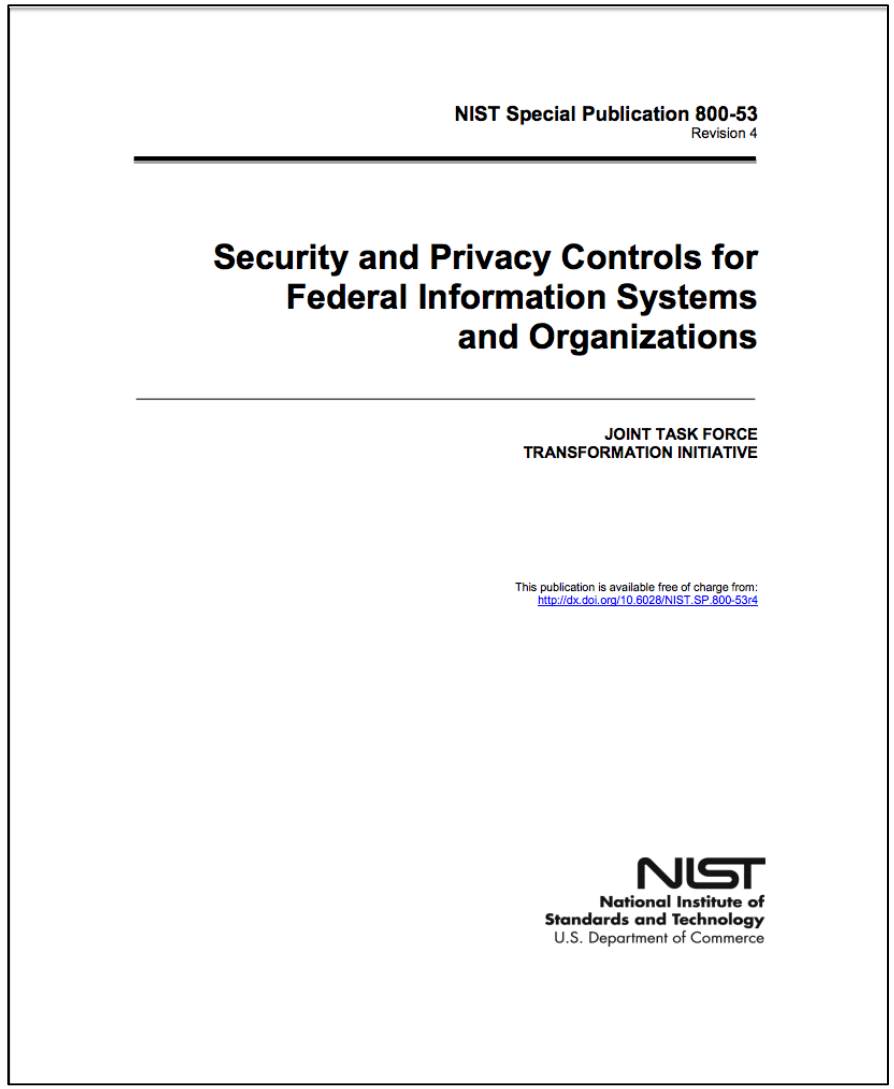
DISA STIGs
The Defense Information Systems Agency's Security Technical Implementation Guides

FIPS
The Federal Information Processing Standards

FedRAMP
The Federal Risk and Authorization Management Program

CIS Benchmarks
The Center for Internet Security

Understanding, documenting, and implementing all that is required can be ***overwhelming*** and ***difficult***



NIST Special Publication 800-53
Revision 4

Security and Privacy Controls for Federal Information Systems and Organizations

JOINT TASK FORCE
TRANSFORMATION INITIATIVE

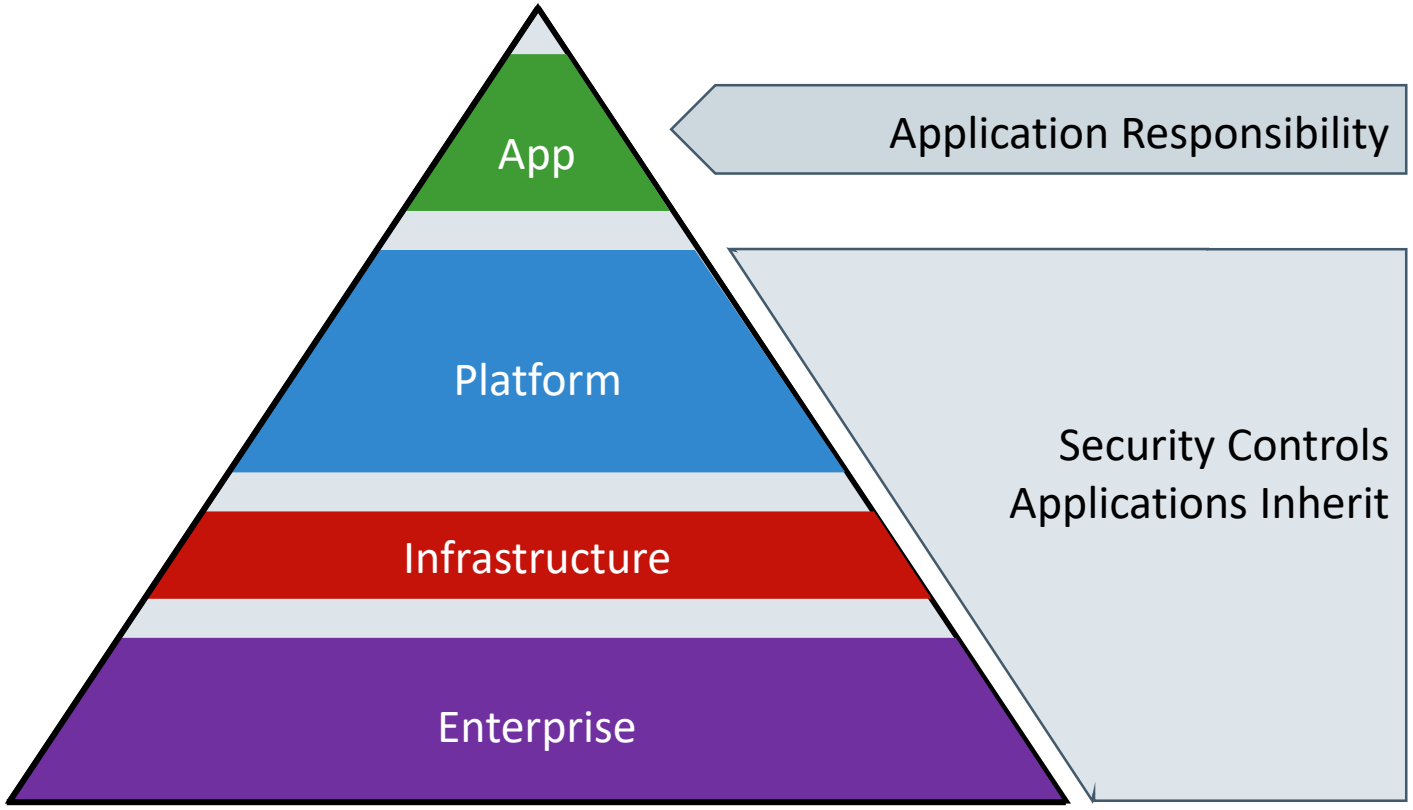
This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.800-53r4

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

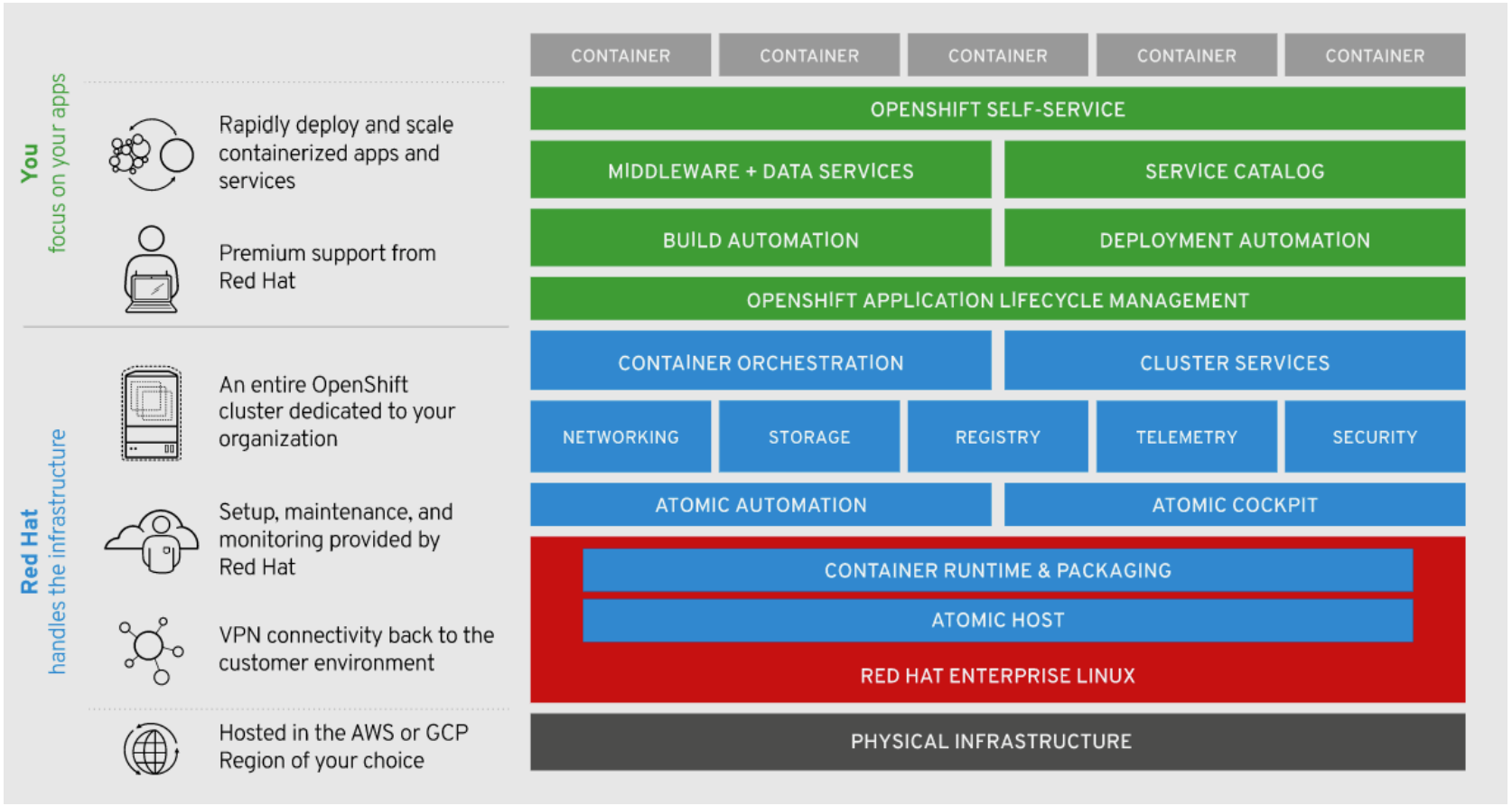
Page 1 of **462**

MORE CHALLENGES STILL

When modernizing legacy applications Cloud Infrastructure and Containerization add new layers of abstraction to the stack, each requiring their own security controls.

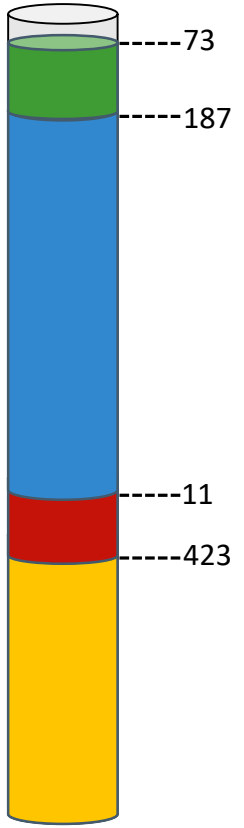


OPENSHIFT IS THE ENTERPRISE GRADE KUBERNETES RUN ON REDHAT ENTERPRISE LINUX



from https://www.openshift.com/dedicated/index.html

Security Controls



Tenant responsibility

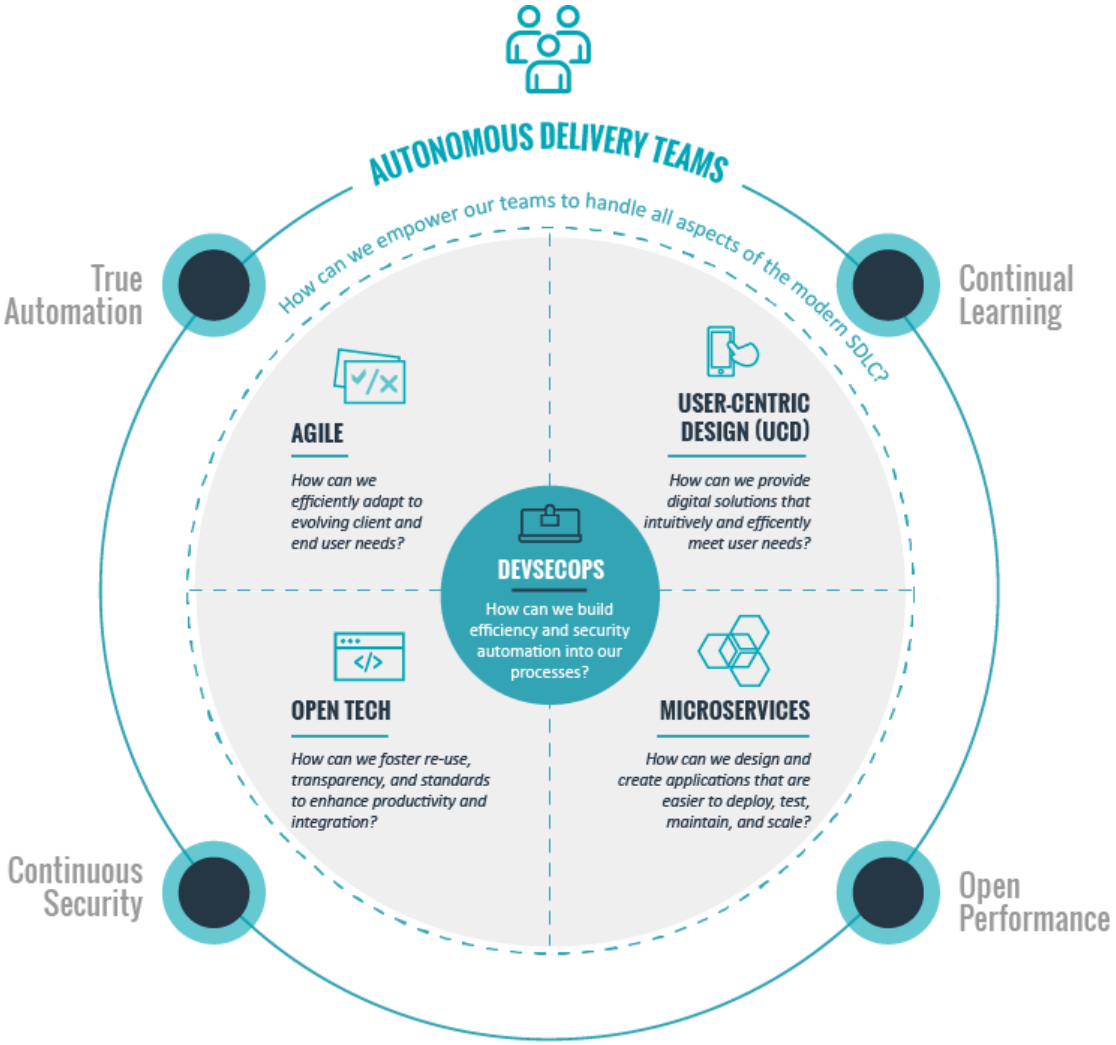
Controls inherited from OpenShift cluster.

Controls inherited from physical infrastructure provider

Controls typically inherited from parent organization.

OpenShift Compliance Guide

BOOZ ALLEN'S UNIFIED MODERN SD APPROACH



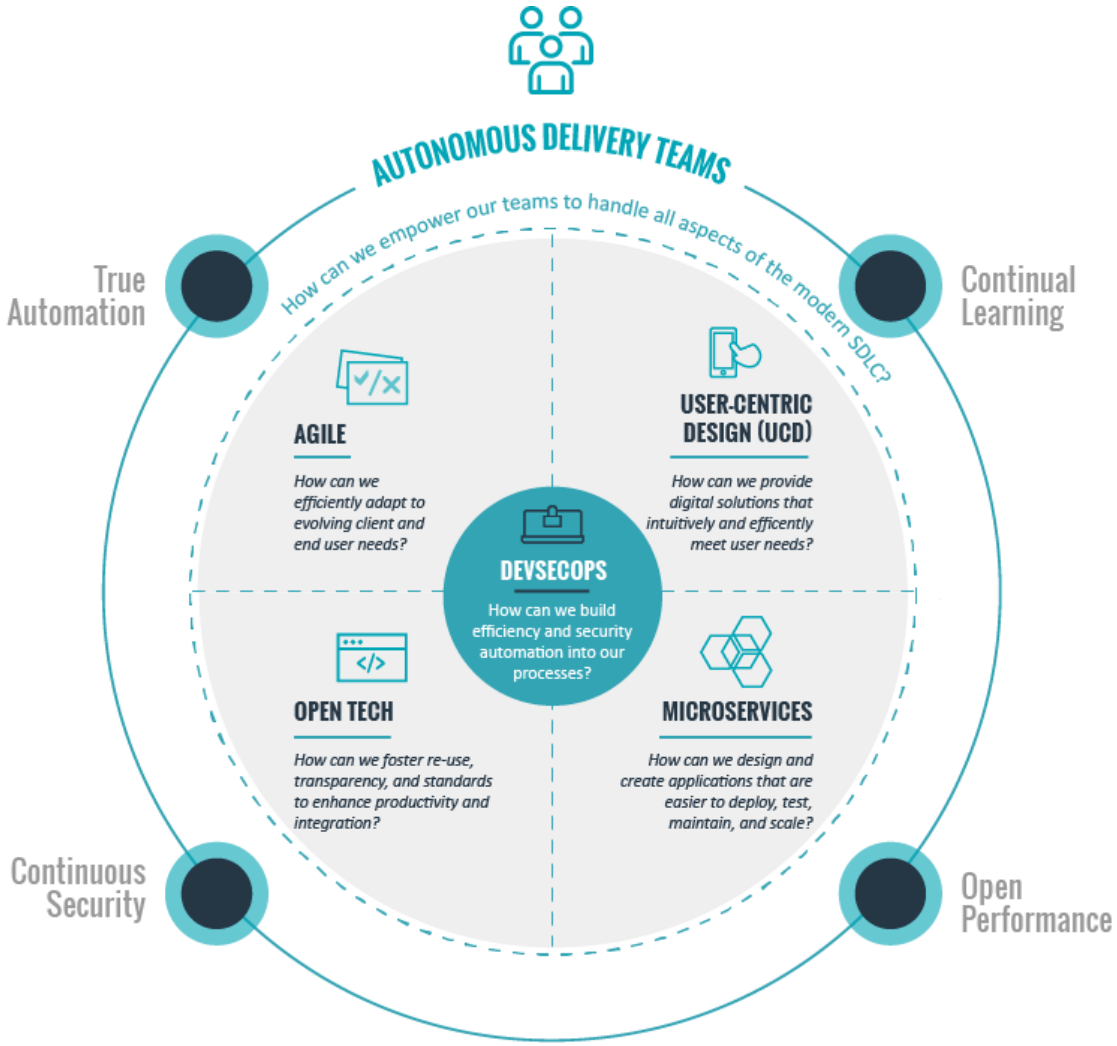
4 CORE TENETS

- Open Performance
- Continuous Security
- True Automation
- Continual Learning

6 INTEGRATED CAPABILITIES

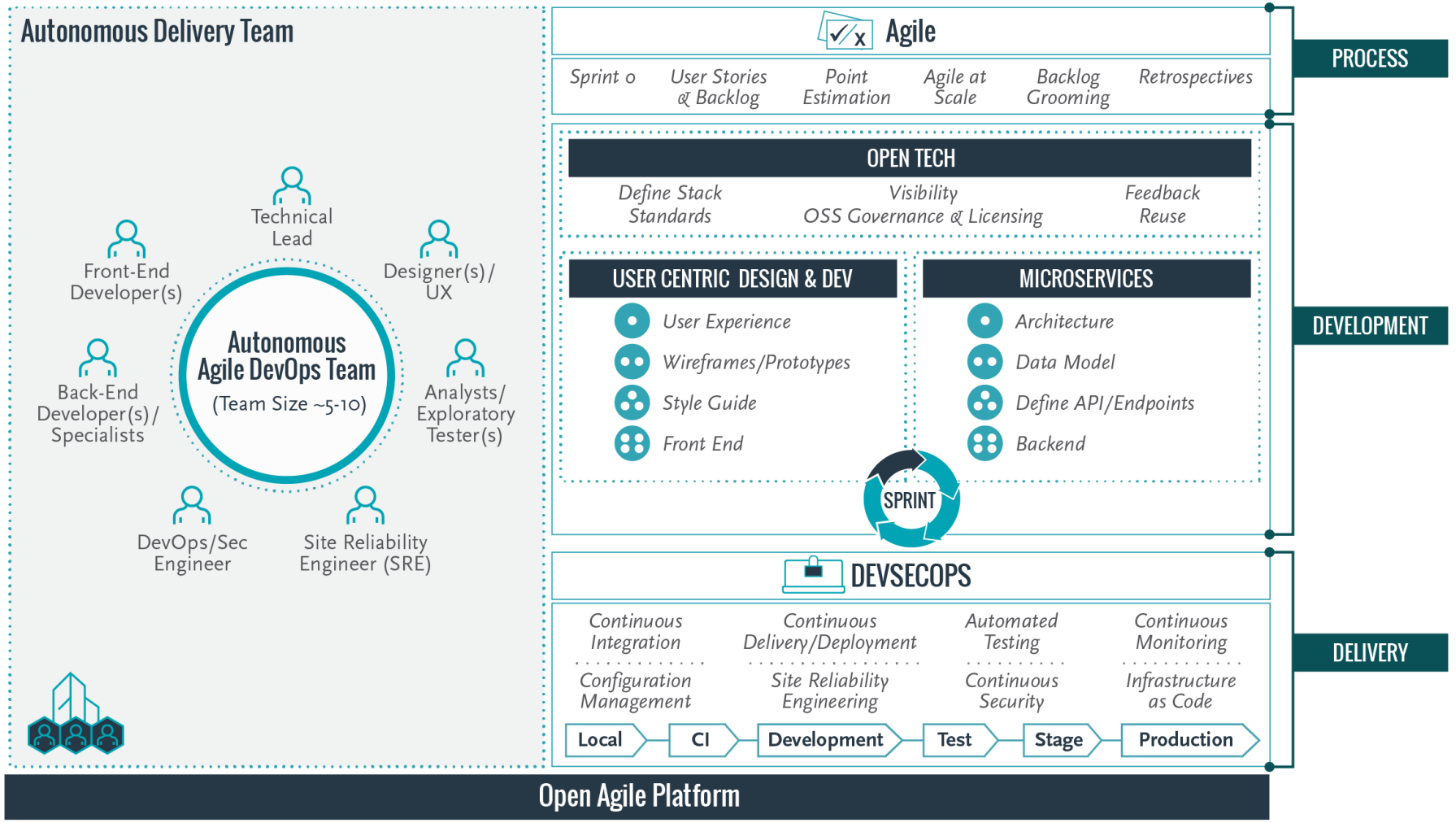
- Autonomous Delivery Teams
- Agile
- User-Centric Design (UCD)
- Open Tech
- Microservices
- DevSecOps

BOOZ ALLEN'S UNIFIED MODERN SD APPROACH



- Small Fully Integrated Autonomous Delivery Teams
- Team Owns All Aspects from Dev to Ops
- Shorten time for Resolution of Issues & Delivery of Features
- Guarantee Stable and Repeatable Operating Environments Every Time
- Automate as much as Possible (test, infra, deploy, etc.)
- Proactively Stop and Fix Potential Defects
- Transparency & Continuous feedback
- Shift Security to the left and throughout
- Focus On and Validate User Experiences
- Granular Services and Functions

BOOZ ALLEN'S UNIFIED MODERN SD APPROACH



[This Area Represents the Common Platform, Services, Environment, Infrastructure - Customize to meet specific client/reponse requirements]

BOOZ ALLEN DEVSECOPS PUTS THE SECURITY IN DEVOPS

CONTINUOUS SECURITY & COMPLIANCE IS PERVASIVE IN OUR DEVOPS APPROACH. IT CROSS-CUTS EVERY PRACTICE AREA

Security and compliance are indicative of the same software delivery sins that spawned the DevOps movement. Work piles up because it is tedious, foreign, or difficult. Security pros are alienated and left to burn down the pile in isolation, as an afterthought. True concerns then become hugely disruptive, which breeds further discontent within the team.

AS WITH QUALITY ASSURANCE, SECURITY ASSURANCE AND COMPLIANCE CAN BE INTEGRATED INTO YOUR SOFTWARE DEVELOPMENT LIFECYCLE

- **Shift-left** many security and compliance activities as a **shared responsibility** of the whole team.
- **Educate and automate** security vigilance to establish **early detection, confidence, and trust** required for Continuous Delivery.
- Perform vulnerability and compliance **inspection** of dependencies, code, container images, and running applications

Dependencies

Prevent introduction of vulnerabilities from the outside. Scan libraries in dependency repos, source code repos, and on disk for known vulnerabilities.

Static Code Analysis

Analyze the code written by developers for inadvertent technical and logical flaws that make it vulnerable.

Dynamic Application Security Testing

Perform automated penetration testing to see how your application will withstand common attacks at runtime.

Image Scanning

Unpack and scan dependencies and configuration of the image to be used at runtime for vulnerabilities, out-of-date patching, and to ensure a trusted pedigree.

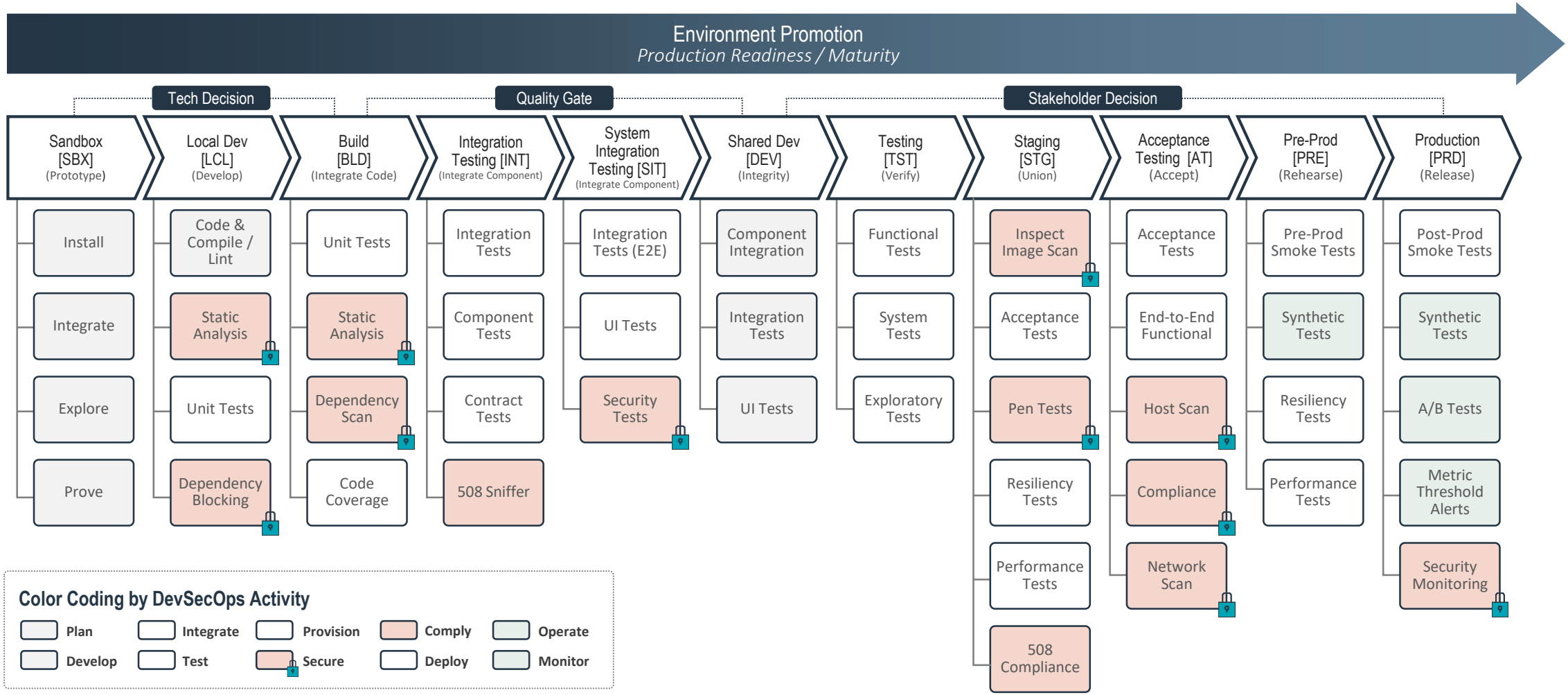
Continuous Compliance

Routinely scan the configuration of hosts or containers in their packaged image state or at runtime for compliance with security policy groups (NIST, CIS, FISMA, STIG, etc.), for required patches, or for configuration drift.

Accessibility Assurance

Crawl web pages for compliance with section 508 standards to give developers early warning and opportunity to improve the site while accelerating manual 508 testing.

THE DEVSECOPS PIPELINE IS A CONFIGURABLE WORKFLOW



Note: Not All Environment Types are Required

DEVSECOPS: YOU'RE NOT DONE ONCE YOU'RE IN PRODUCTION

Continuously monitor containers in production for security policy violations

Stay Secure with

- **Capture the activity on both sides of the security event**
- **Log every system call on your cluster**
- **Pause containers who violate security policies**
- **Alert teams when policy breaches occur**

THE BAH SOLUTIONS DELIVERY PLATFORM (SDP) IS THE PURPOSE-BUILT SOLUTION THAT ENABLES OUR PHILOSOPHY AND PUTS PROCESSES AND PRACTICES INTO ACTION

REFERENCE MODEL FOR THE SOLUTIONS DELIVERY PLATFORM

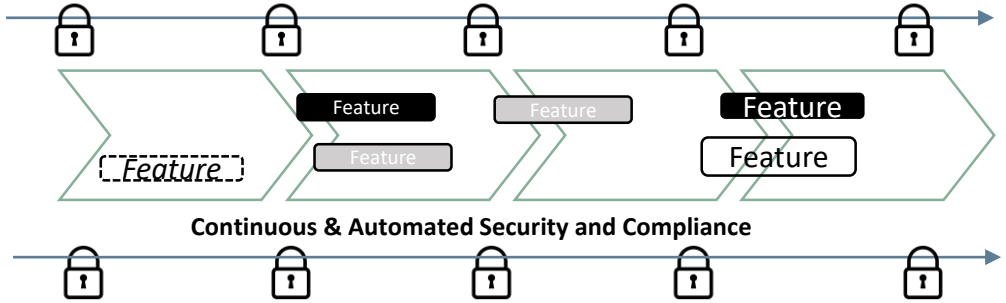
DevSecOps Dashboard

Aggregate metrics for actionable insights to achieve continuous learning

Secure Pipeline

Standardize your organization's continuous delivery

Automated provenance with a trusted supply chain to production



Container Platform

Support all the activity

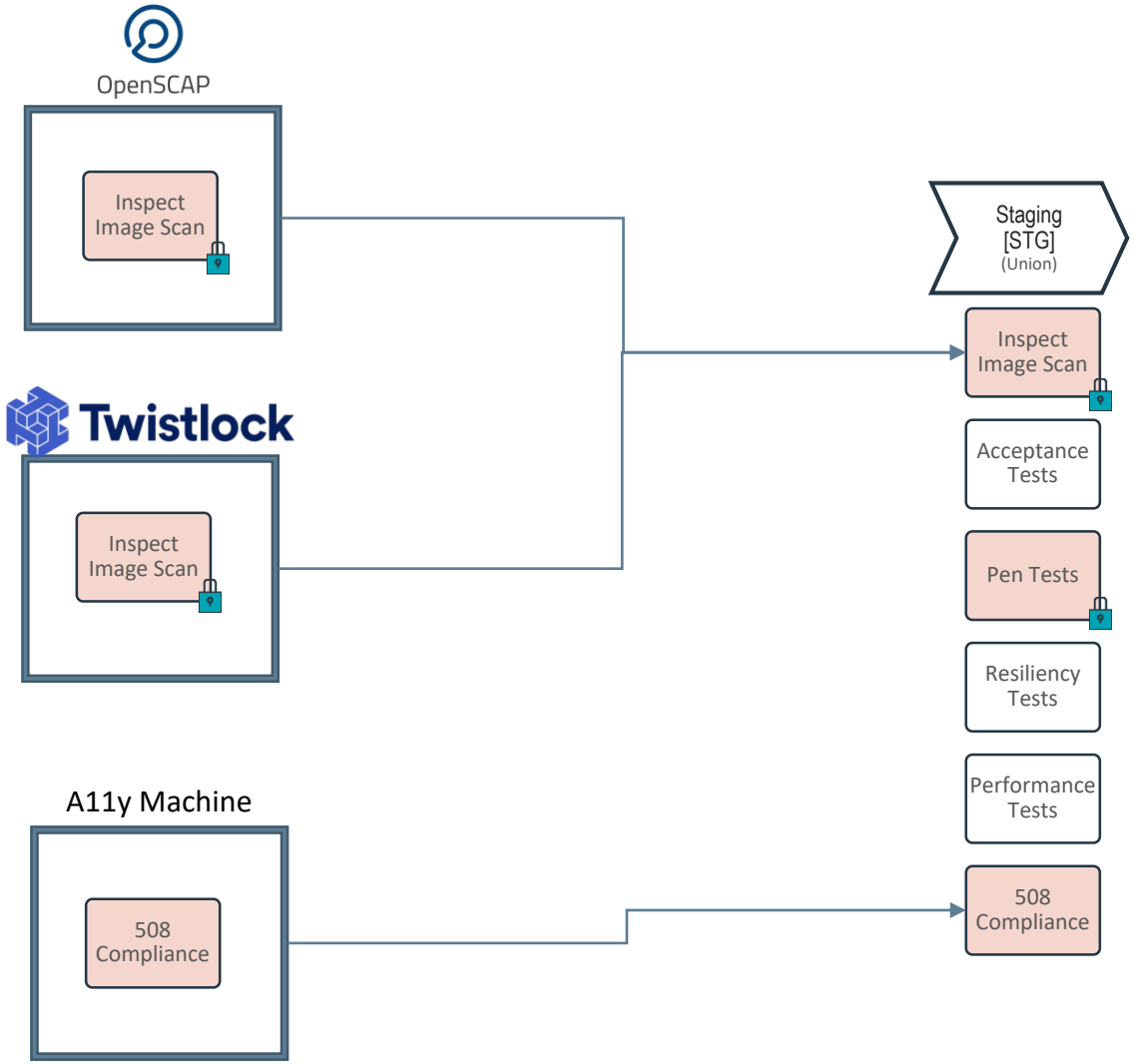
BENEFITS

- End-to-end **traceability** of delivery
- Real-time **status** at a glance
- **Single view** of multiple apps/components and teams
- High-fidelity **drill-down** to activity-specific metrics

- **Automate** delivery, and assurance of security & quality
- Enable **secure, on-demand** flow of new features
- Continuous, quantitative, and actionable **feedback**
- **Shifting-left security** and streamlining activities **mitigates risk** by avoiding big and long releases

- **Productivity** increase with self-service, homogenous IT
- **Scalable, resilient** backbone
- Environment **parity**
- Improved resource **utilization**

SOLUTIONS DELIVERY PLATFORM PIPELINE FRAMEWORK: WHAT?

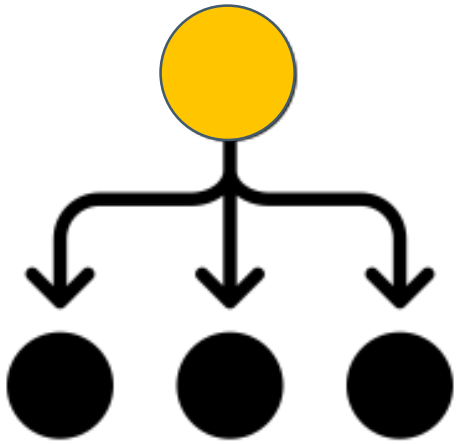


Shared Libraries Abstract Reusable Functionality

Allows composable “plug and play” pipelines

SOLUTIONS DELIVERY PLATFORM PIPELINE FRAMEWORK: WHAT?

A Single Organizational Jenkins Pipeline



ORGANIZATION CONFIGURATION FILE

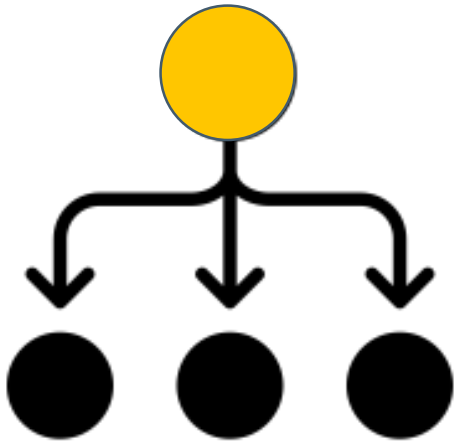
```
use_pipeline_template = true

sdp_image_repository = ”…"
sdp_image_repository_credential = ”…"

application_image_repository = ”…"
application_image_repository_credential = ”…"

application_environments{
  dev{
    short_name = "dev"
    long_name = "Development"
  }
  prod{
    short_name = "prod"
    long_name = "Production"
  }
}

stages{
  continuous_integration{
    unit_test
    static_code_analysis
    build
    scan_container_image
  }
}
```

```
libraries{
  github_enterprise
  sonarqube{
    enforce_quality_gate = true
  }
  docker
  twistlock{
    url = ”…"
    credential = ”…"
  }
  openshift{
    url = “…"
    helm_configuration_repository = ”…"
    helm_configuration_repository_credential = ”…”
    tiller_namespace = ”…"
    tiller_credential = ”…"
  }
  owasp_zap{
    merge = true
    vulnerability_threshold = "High"
  }
  slack
}

notifiers{
  slack
}
```

*specifies pipeline composition for the agency and determines which configurations a tenant can override

SOLUTIONS DELIVERY PLATFORM PIPELINE FRAMEWORK: WHAT?

A Single Organizational Jenkins Pipeline



ORGANIZATION JENKINSFILE (PIPELINE AS CODE)

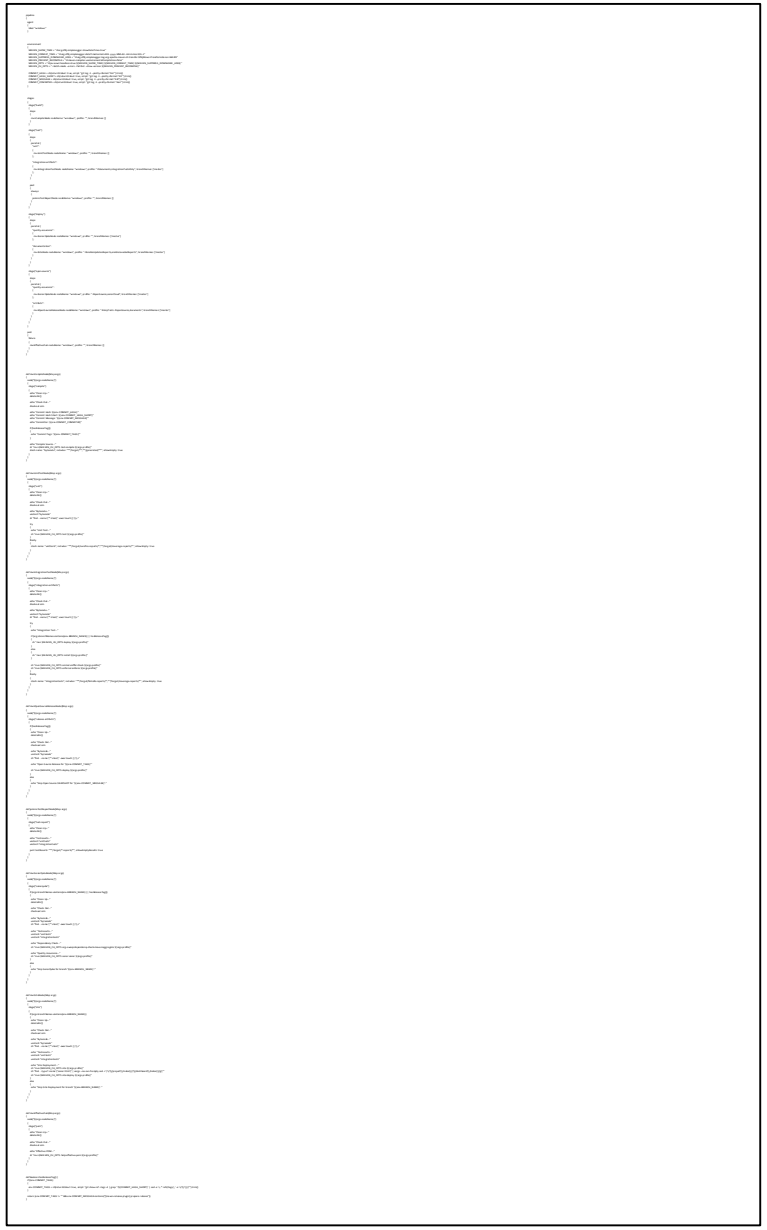
```
on_commit{
  continuous_integration()
}

on_pull_request to: master, from: feature, {
  continuous_integration()
  deploy_to preview
  parallel “Penetration Test”: { penetration_test() },
          “Accessibility Test”: { accessibility_compliance_test() },
          “Performance Test”: { performance_test() },
          “Functional Test”: { functional_test() }
}

on_merge to: master, {
  deploy_to production
}
```

COMPARE AND CONTRAST

ORGANIZATION JENKINSFILE (PIPELINE AS CODE)

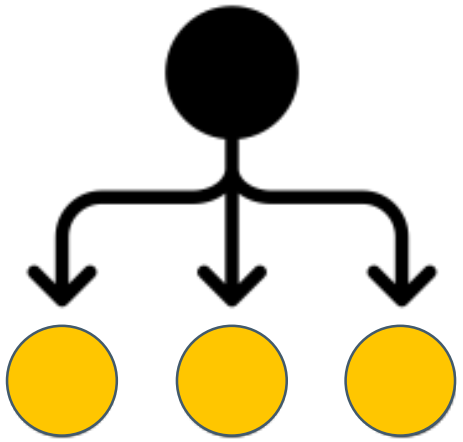
```
on_commit{
  continuous_integration()
}

on_pull_request to: master, from: feature, {
  continuous_integration()
  deploy_to preview
  parallel “Penetration Test”:   { penetration_test() },
          “Accessibility Test”: { accessibility_compliance_test() },
          “Performance Test”:   { performance_test() },
          “Functional Test”:    { functional_test()  }
}

on_merge to: master, {
  deploy_to production
}
```

TYPICAL JENKINSFILE



SOLUTIONS DELIVERY PLATFORM PIPELINE FRAMEWORK: WHAT?

A Single Organizational Jenkins Pipeline



TENANT CONFIGURATION FILE

```
libraries{
  owasp_zap{
    target = “https://example.com”
  }
}
steps{
  unit_test{
    image = “maven”
    command = “mvn clean verify”
  }
}
```

SOLUTIONS DELIVERY PLATFORM PIPELINE FRAMEWORK: WHAT?

PUTTING IT ALL TOGETHER

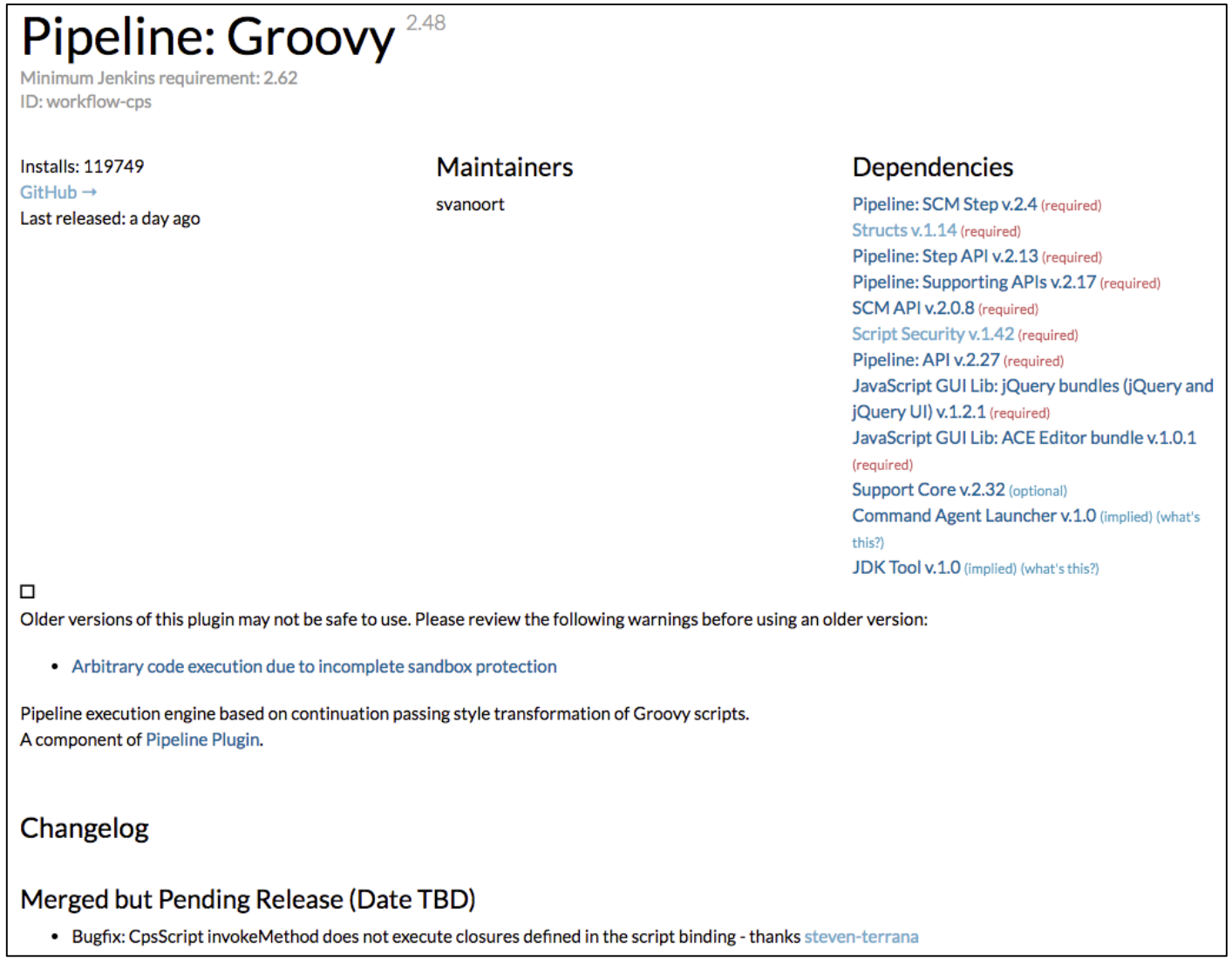
ORGANIZATION CONFIGURATION FILE	TENANT CONFIGURATION FILE	SHARED LIBRARIES	SINGLE DEVSECOPS PIPELINE
Govern	**Define**	**Compose**	**Execute**

SOLUTIONS DELIVERY PLATFORM PIPELINE FRAMEWORK: HOW?

2 Differentiators

- **Contributed back to the Jenkins project with a bug fix enabling more dynamic behavior**
  - https://github.com/jenkinsci/workflow-cps-plugin/pull/204

- **Modifications to the Pipline Multibranch: with Defaults plugin enabling the use of a single Jenkinsfile across an entire GitHub Organiization**
  - Contribution back to open source pending



Pipeline: Groovy 2.48

Minimum Jenkins requirement: 2.62
ID: workflow-cps

Installs: 119749
GitHub →
Last released: a day ago

Maintainers

svanoort

Dependencies

Pipeline: SCM Step v.2.4 (required)
Structs v.1.14 (required)
Pipeline: Step API v.2.13 (required)
Pipeline: Supporting APIs v.2.17 (required)
SCM API v.2.0.8 (required)
Script Security v.1.42 (required)
Pipeline: API v.2.27 (required)
JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) v.1.2.1 (required)
JavaScript GUI Lib: ACE Editor bundle v.1.0.1 (required)
Support Core v.2.32 (optional)
Command Agent Launcher v.1.0 (implied) (what's this?)
JDK Tool v.1.0 (implied) (what's this?)

Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- Arbitrary code execution due to incomplete sandbox protection

Pipeline execution engine based on continuation passing style transformation of Groovy scripts.
A component of Pipeline Plugin.

Changelog

Merged but Pending Release (Date TBD)

- Bugfix: CpsScript invokeMethod does not execute closures defined in the script binding - thanks steven-terrana

SA-11 - Developer Security Testing And Evaluation

Requirement: DEVELOPER SECURITY TESTING AND EVALUATION Control: The organization requires the developer of the information system, system component, or information system service to: a. Create and implement a security assessment plan; b. Perform [Selection (one or more): unit; integration; system; regression] testing/evaluation at [Assignment: organization-defined depth and coverage]; c. Produce evidence of the execution of the security assessment plan and the results of the security testing/evaluation; d. Implement a verifiable flaw remediation process; and e. Correct flaws identified during security testing/evaluation.

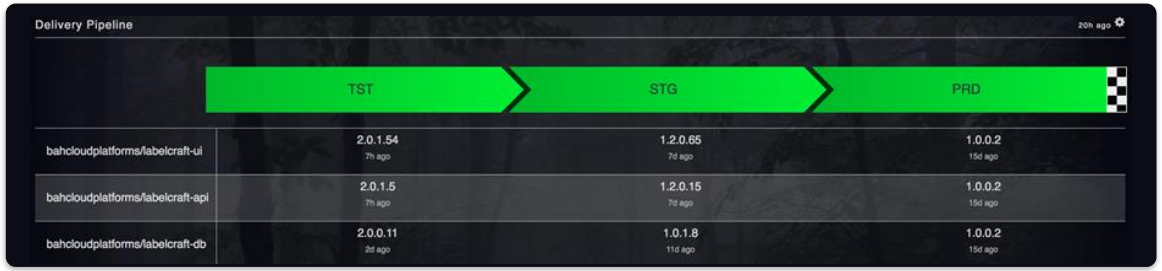
SA-3 - System Development Life Cycle

Requirement: SYSTEM DEVELOPMENT LIFE CYCLE Control: The organization: a. Manages the information system using [Assignment: organization-defined system development life cycle] that incorporates information security considerations; b. Defines and documents information security roles and responsibilities throughout the system development life cycle; c. Identifies individuals having information security roles and responsibilities; and d. Integrates the organizational information security risk management process into system development life cycle activities.

THE HYGIEIA DASHBOARD PROVIDES METRICS AND VISIBILITY INTO THE EFFECTIVENESS OF THE PROCESS, ENVIRONMENTS, AND OPERATIONS

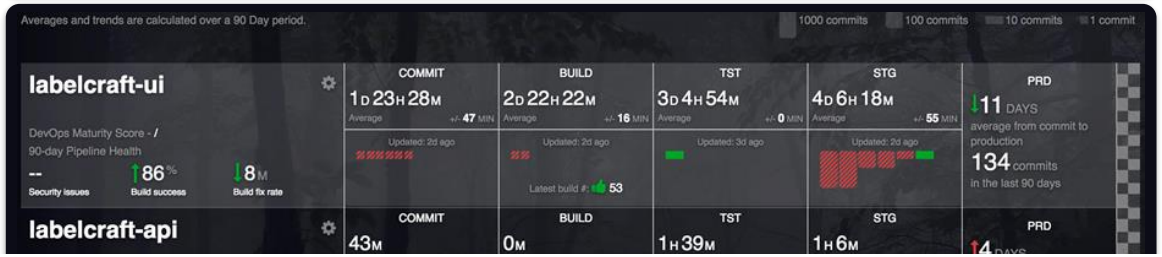
Deployments: *Feedback on each deployment*

- What artifacts, at what version, are running where



Value Stream Flow *Feedback on pace and health of multiple-team delivery*

- How close to delivering the next version
- Is delivery speeding up or slowing down
- At what stage(s) are the bottlenecks



Team Dashboard

Provides deep measures across the end-to-end app delivery lifecycle

- Feature Backlog status,
- code change activity,
- current code quality,
- build success,
- environment deployments

Dashboard can be configured to integrate organizationally specific tools and to develop custom new features

WE’RE HIRING -- BOOZALLEN.COM/APPLY