

RSA[®]Conference2019

San Francisco | March 4–8 | Moscone Center



BETTER.

SESSION ID: ASD-W02

Will your application be secure enough when Robots produce code for you?

Hasan Yasar

Technical Manager, Faculty Member

SEI – CMU

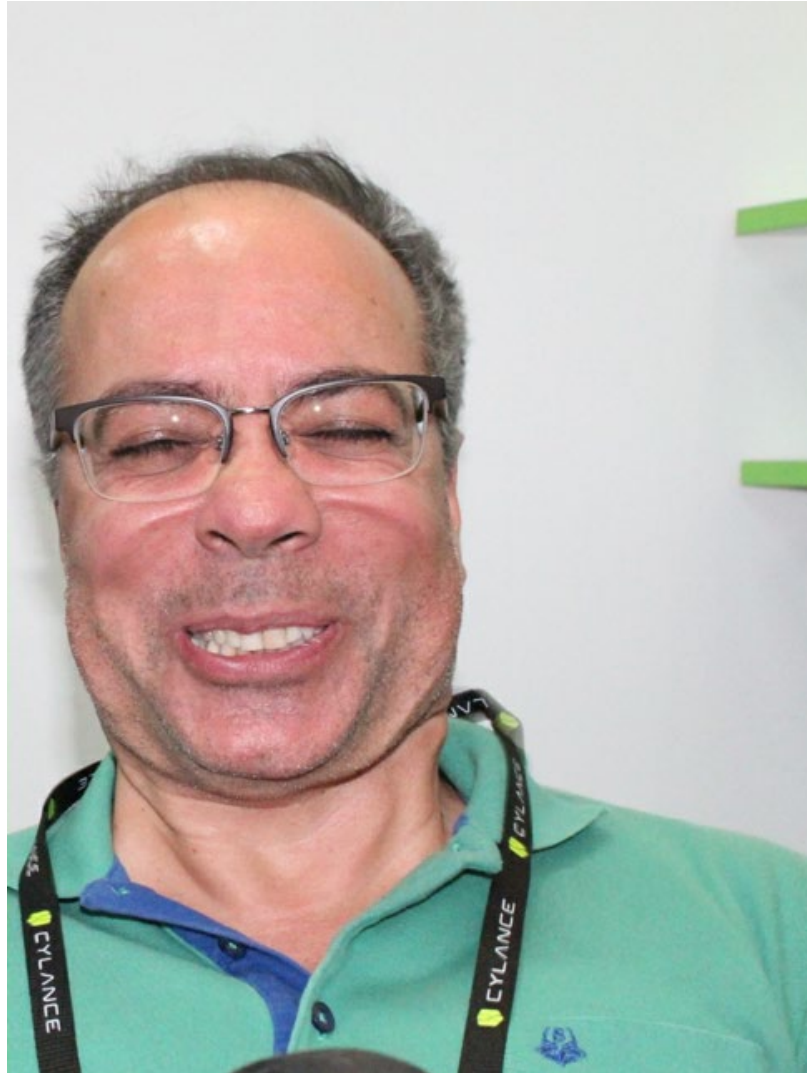
@securelifecycle



#RSAC

With the speed of DevOps...

It is me!
I felt the speed of
DevOps



The future is here
with
DevSecOps

Agenda

- Why matters?
- Current Landscape
- Security Framework: What IA wants?
- The DevSecOps Factory
- Be Ready...



RSA[®]Conference2019

Why Matters?





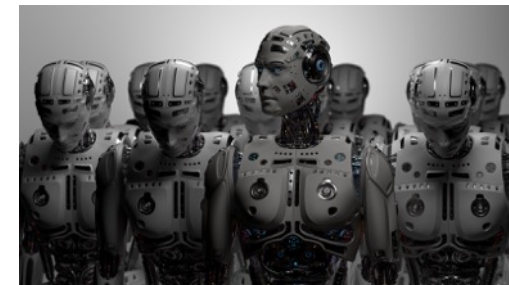
Automation

- DevOps is everywhere
- Increase of using
 - Automation
 - Open Source libraries
 - Containerization
 - Re-use deployment / IaC scripts



New attack vectors

- Taking advantage of automation
- Open source libraries
- Fast build-deploy
- Containerization
- Complexity and dependencies



Recap of a few recent data breaches

- Google+ blunder exposed Data from 52.5 million
- Marriott :Data on 500 million guests stolen in 4 year breach
- Compromised NPM packages
- GDPR: 8,000 Data Breach Reports Filed So Far in UKZ



A Shifting Battlefield of Attacks: Hackers Inject Malicious Code into Supply Chains

March 2016 - August 2018

1

left-pad: Popular npm packages were removed from the repository, breaking thousands of websites and revealing how changes can immediately propagate to the real world.

2

npm credentials used by publishers of 79,000 packages were published online or easily compromised by dictionary attacks.

3

npm typosquat: 40 intentionally malicious packages harvested credentials used to publish to the npm repository itself.

4

docker123321 images were created on Docker Hub. In Jan'18, it was accused of poisoning a Kubernetes honeypot, then in May'18 it was equated to a crypto mining botnet.

5

PyPI typosquat: The Slovak National Security Office (NBU) found 10 malicious PyPI packages. Evidence of the fake packages being downloaded and incorporated into software multiple times was noted between Jun'17 and Sept'17.

6

Malicious npm: Gilbertson writes a fictional tale of creating a malicious npm packages to harvest credit card numbers from hundreds of websites.

7

npm credentials: A core contributor to the conventional-changelog ecosystem had their npm credentials compromised and a malicious version of the package was published. Package was installed 28,000 times in 35 hours and executed a Monero crypto miner.

go-bindata: after a developer deleted their GitHub account, someone immediately grabbed the ID — inheriting the karma instilled in that id, calling into question what packages and sources are canonical and immutable.

8

9

Backdoored npm: The npm security team responded to reports of a package that masqueraded as a cookie parsing library but contained a malicious backdoor. Published in March'18 to introduce unauthorized publishing of mailparser; despite being deprecated, mailparser still received about 64,000 weekly downloads.

10

Backdoored PyPI: SSH Decorator (ssh-decorate), a library for handling SSH connections from Python code, was backdoored to enable stealing of private SSH credentials.

11

homebrew breach: Eric Holmes, an operations engineer at Remind, gained commit access to homebrew in under 30 minutes through an exposed GitHub API token. While he had no malicious intent, he gained access to components that are downloaded 500,000 times per month.

Mar 2016

July 2017

Sep 2017

Jan 2018

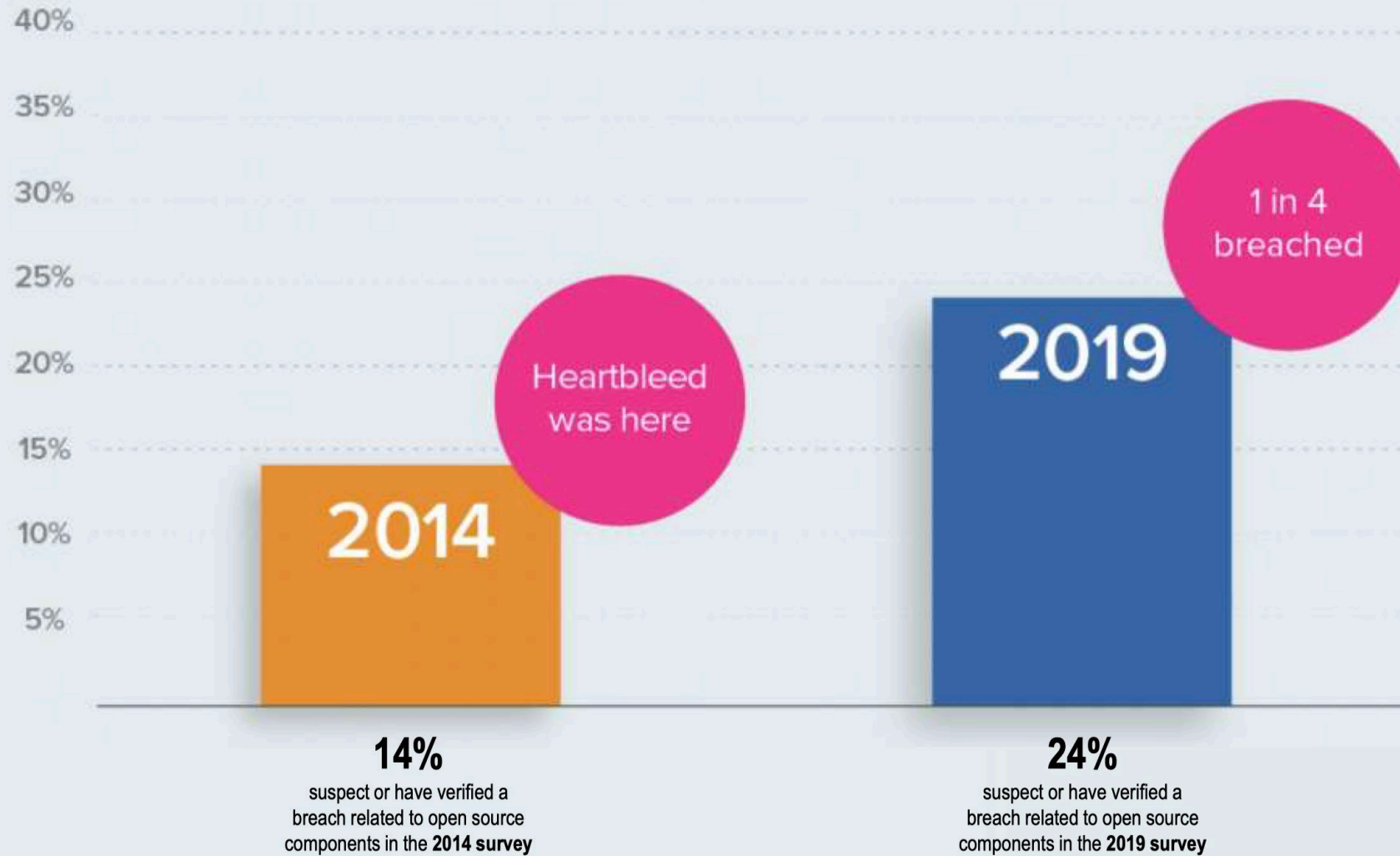
Feb 2018

May 2018

Aug 2018

* 2018 State of the Software Supply Chain Report

Breaches increased 71%



*DevSecOps Community
Survey 2014 and 2019

RSA®Conference2019

Current Landscape

Dev{Sec}Ops



The Foundation: 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 system¹

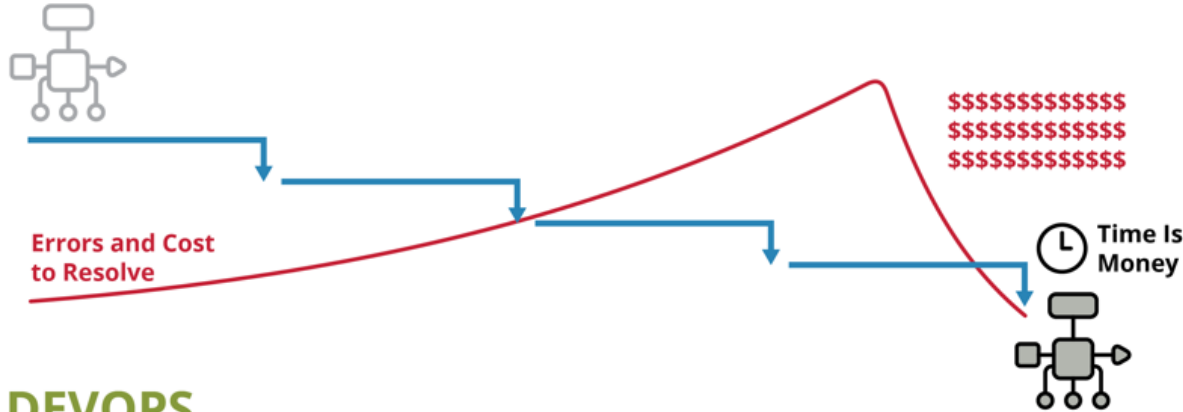
Four Fundamental Principles

1. *Collaboration*: between all stakeholders
2. *Infrastructure as code (IaC)*: assets are versioned, scripted, and shared
3. *Automation*: deployment, testing, provisioning, any manual or human-error-prone process
4. *Monitoring*: any metric in development or operation that can inform priorities, direction, and policy

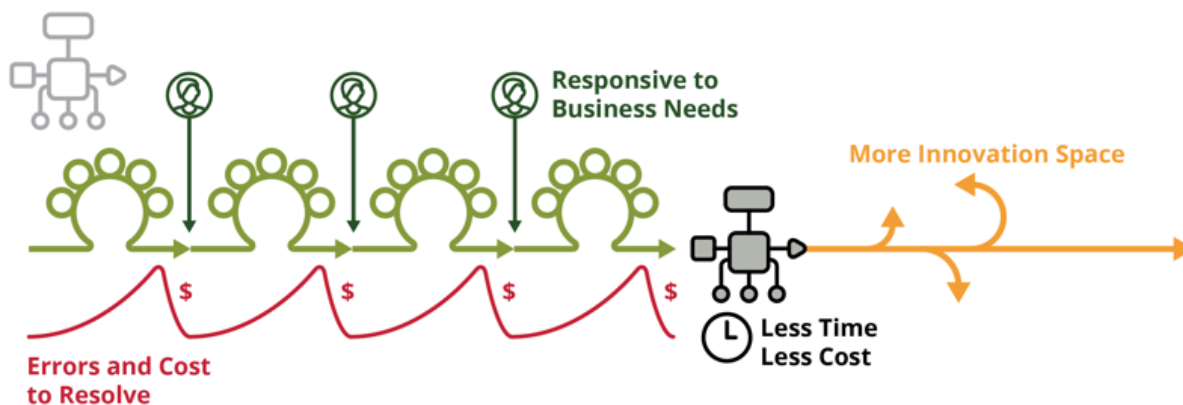
[1] IEEE P2675 DevOps Standard for Building Reliable and Secure Systems Including Application Build, Package, and Deployment

Benefits of DevOps

WATERFALL



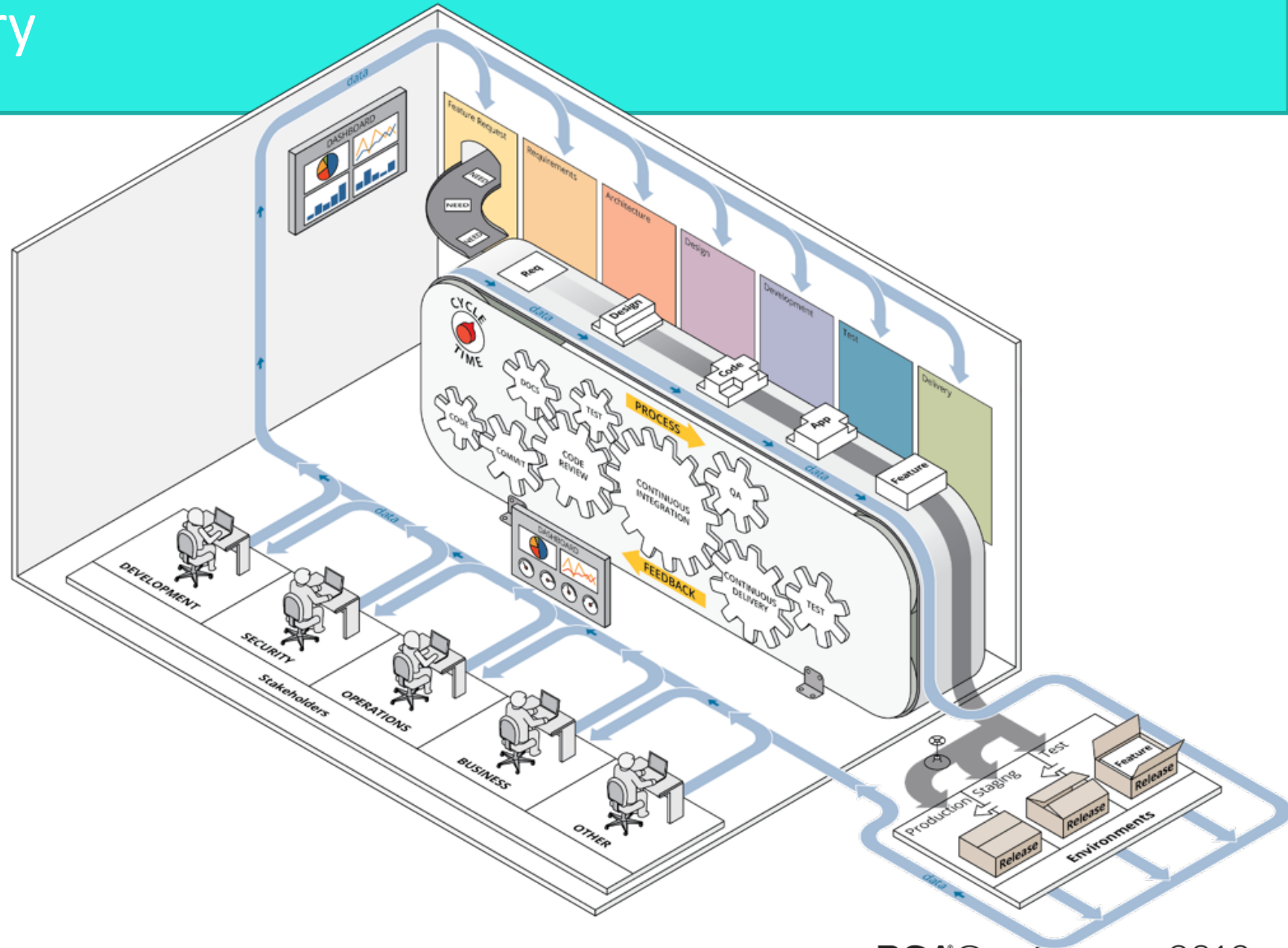
DEVOPS



- 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

The DevOps Factory

- Feature to deployment
- Iterative and incremental development
- Automation in every phase of the SDLC
- Continuous feedback
- Metrics and measurement
- Complete engagement with all stakeholders
- Transparency and traceability across the lifecycle

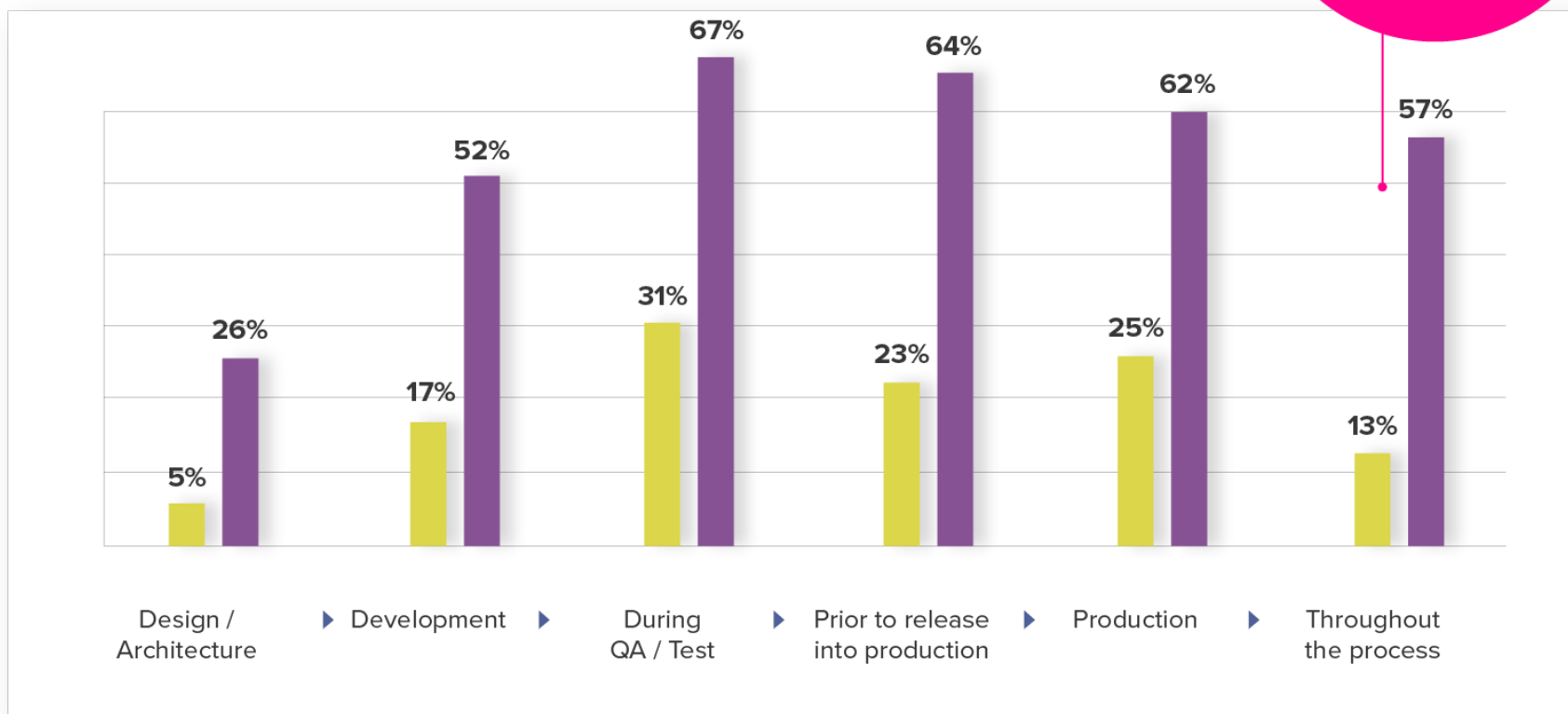


Poll the Audience

- ASD-W02
- Do you have a mature DevOps platform?
 - A. YES
 - B. NO
 - C. Partial
- <https://rsa1-live.eventbase.com/polls?event=rsa2019&polls=3857>

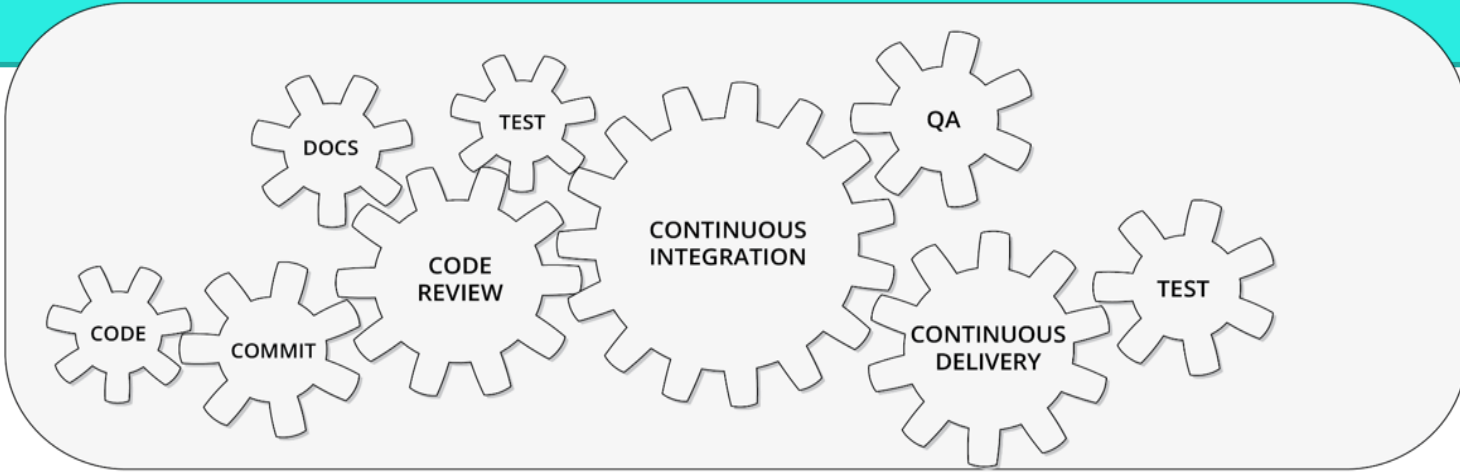
At what point in the development process does your organization perform automated application security analysis?

Mature DevOps practices are 338% more likely to integrate automated security.



Source:
DevSecOps Community
Survey 2018

Security Requires Automation with IaC, CI, and CD



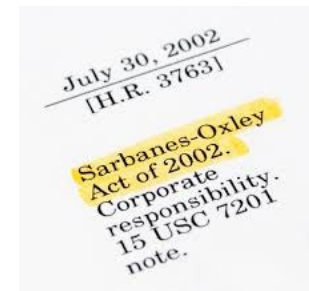
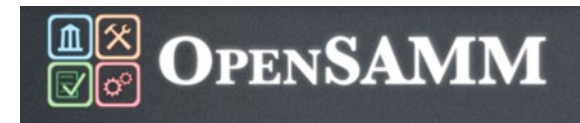
- Security requirements and traceability
 - Risk Management Framework: (1) categorize, (2) select controls, (3) implement, (4) assess, (5) authorize, (6) monitor
- Code review and static analysis
- Automated security testing and verification
- Automated dependency vulnerability analysis
- Immutable system, infrastructure (re-)provisioning

Security Framework: What IA wants?

RMF, ATO &Compliances requirements

Security Framework

- It is time to build security foundation based on..
 - BSIMM
 - OpenSAMM
 - SANS
 - RMF
 - GDPR
 - SOX
 - Open Security Architecture
 -



What is Risk Management Framework (RMF)?

- Information security framework for Authorization to Operate systems
- RMF is a key component of Organization Risk Management
- Explained at NIST Special Publication 800-37
- RMF provides a disciplined and structured process
- Integrates information security and risk management activities into the SDLC

1. Categorize : *information processed, stored, and transmitted*

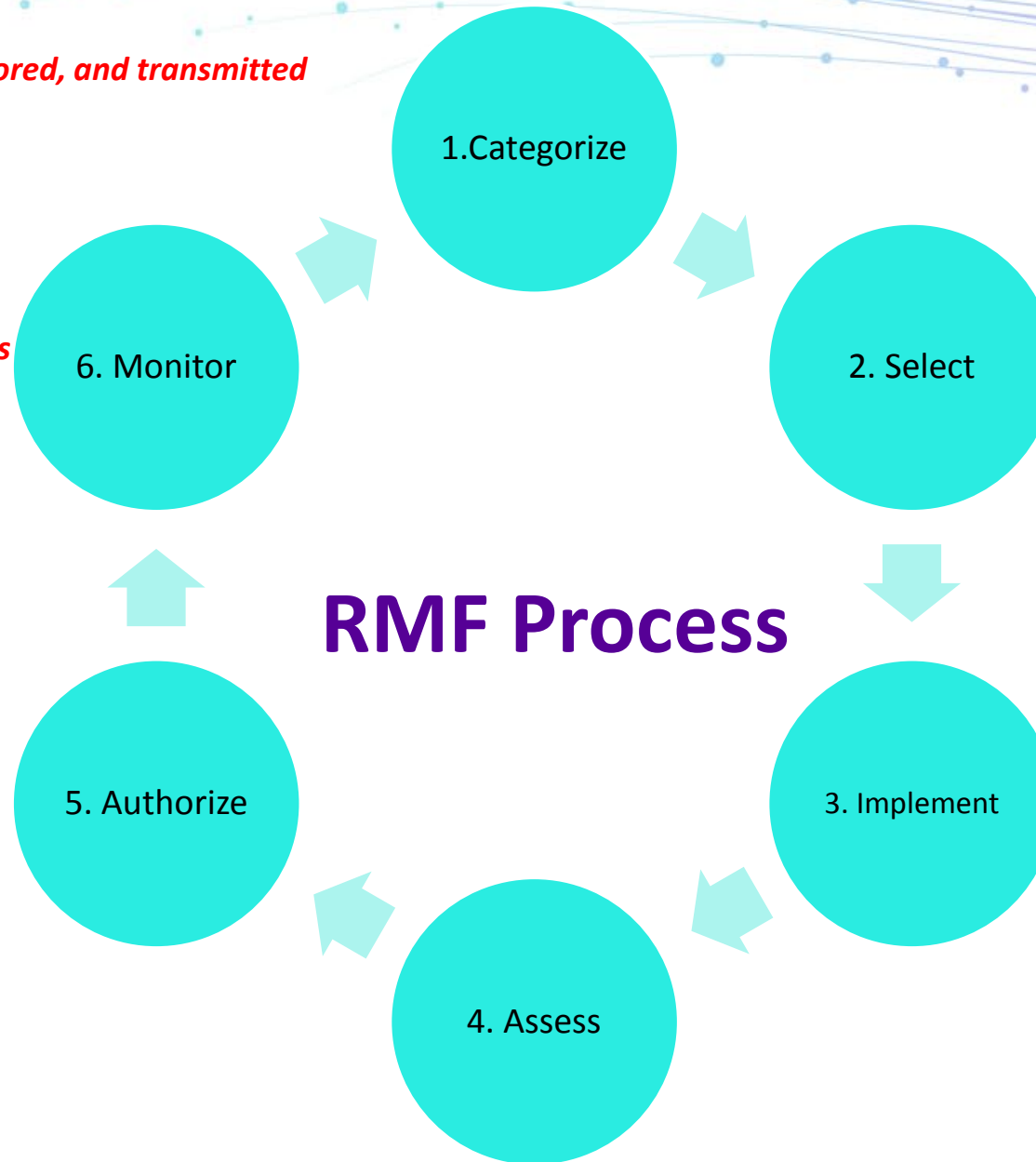
6. Monitor : *ongoing basis including assessing control effectiveness, documenting changes to the system or its environment of operation,*

5. Authorize *determination of the risk to organizational operations and assets, individuals, other organizations*

4. Assess : *controls are implemented correctly,*

2. Select *organizational assessment of risk and local conditions.*

3. Implement : *the controls are employed within the information system and its environment of operation.*



RMF characteristics – NIST 800-37r2

- Real time risk management through the implementation ***continuous monitoring processes***;
- Encourages the use of ***automation*** to make cost-effective, risk-based decisions
- Integrates information security into ***system development life cycle***;
- Provides ***monitoring*** of security controls, and the authorization of information systems;
- Links ***information system level*** to the ***organization level***
- Establishes ***responsibility*** and ***accountability*** for security controls

Compliance, Legal Requirements

- GDPR: General Data Protection Regulation
- FISMA :Federal Information Security Management
- SOX : Sarbanes–Oxley
- HIPAA : Health Insurance Portability and Accountability
- PCI DSS: Payment Card Industry Data Security Standard
- NIST :National Institute of Standards and Technology,
- and others....

Compliance, Legal Requirements

- GDPR: General Data Protection Regulation
- FISMA :Federal Information Security Management
- SOX : Sarbanes–Oxley
- HIPAA : Health Insurance Portability and Accountability
- PCI DSS: Payment Card Industry Data Security Standard
- NIST :National Institute of Standards and Technology,
- And others....

**All requires
Auditing
Traceability
Accountability**

RSA®Conference2019

The DevSecOps Factory

Automation and Security

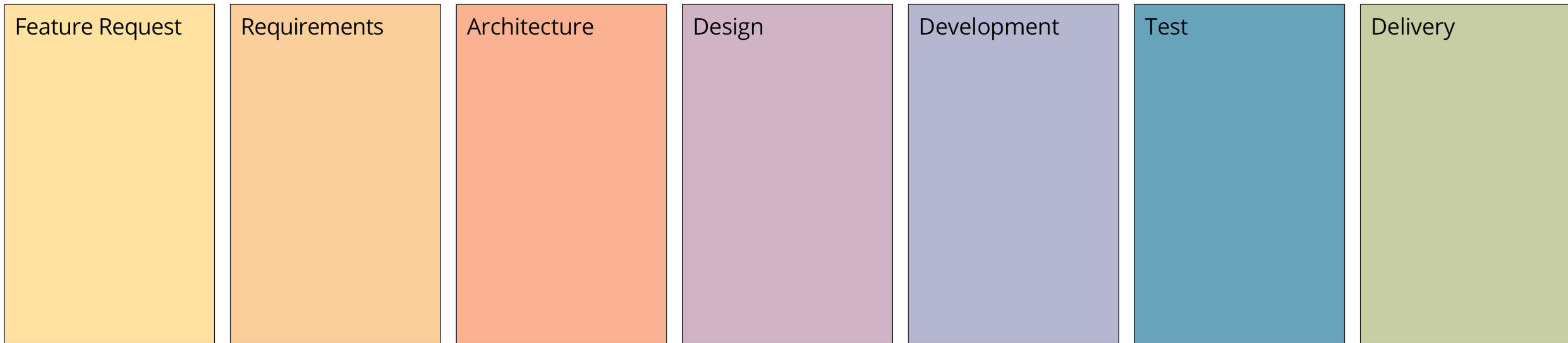
Poll the Audience

- ASD-W02
- Is automation the only way to solve application security?
 - A. YES
 - B. NO
- <https://rsa1-live.eventbase.com/polls?event=rsa2019&polls=3858>

Secure DevOps is a model on integrating the software development and operational process considering security activities: requirements, design, coding, testing

....

DevOps Phases – *on each iteration/sprint*

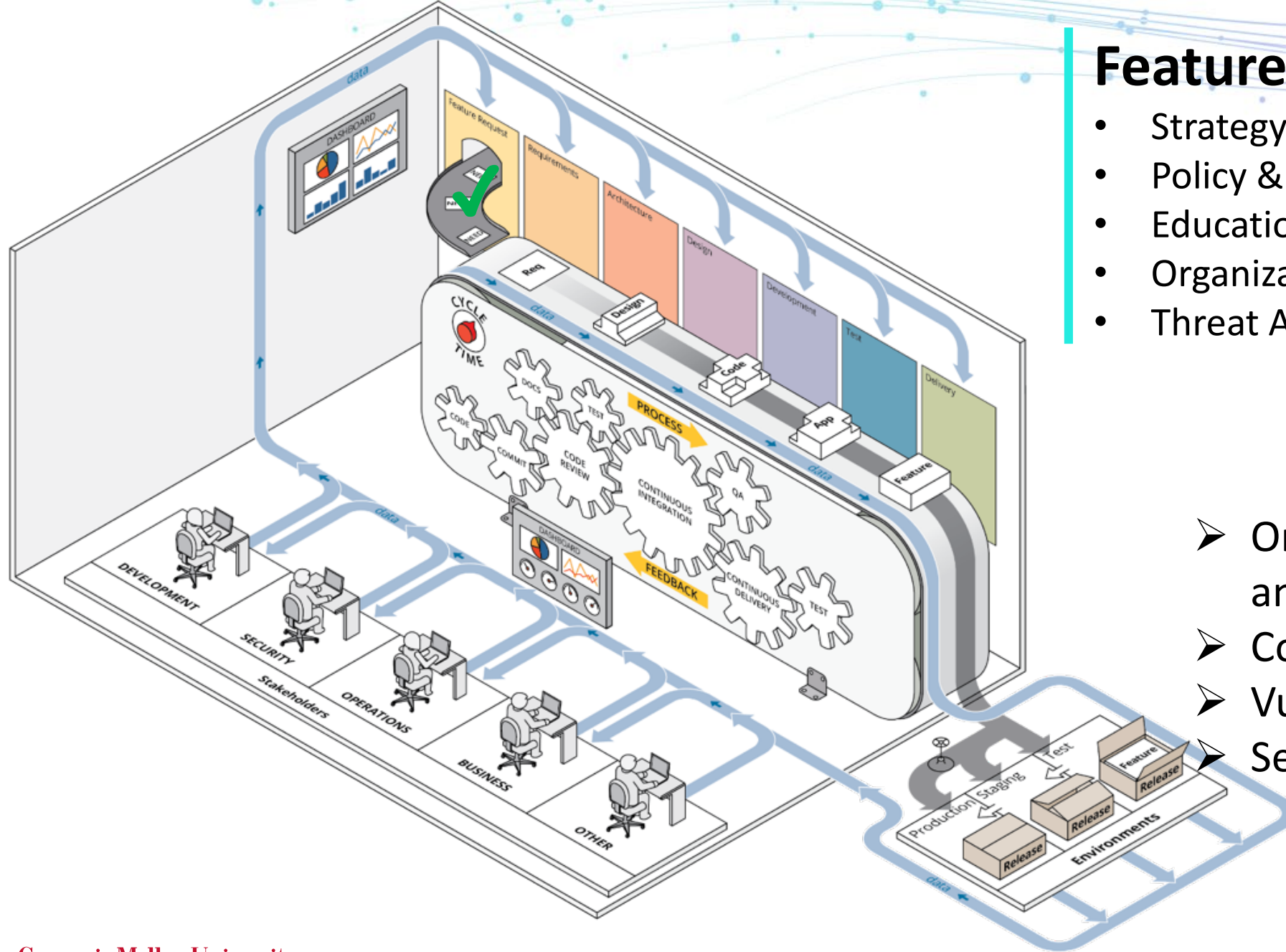


Feature Request

- Strategy & Metrics
- Policy & Governance
- Education & Security Guidance
- Organizational Risk Factors
- Threat Assessment



- Organizational awareness and knowledge
- Common attack vectors
- Vulnerability management
- Security Development Plan

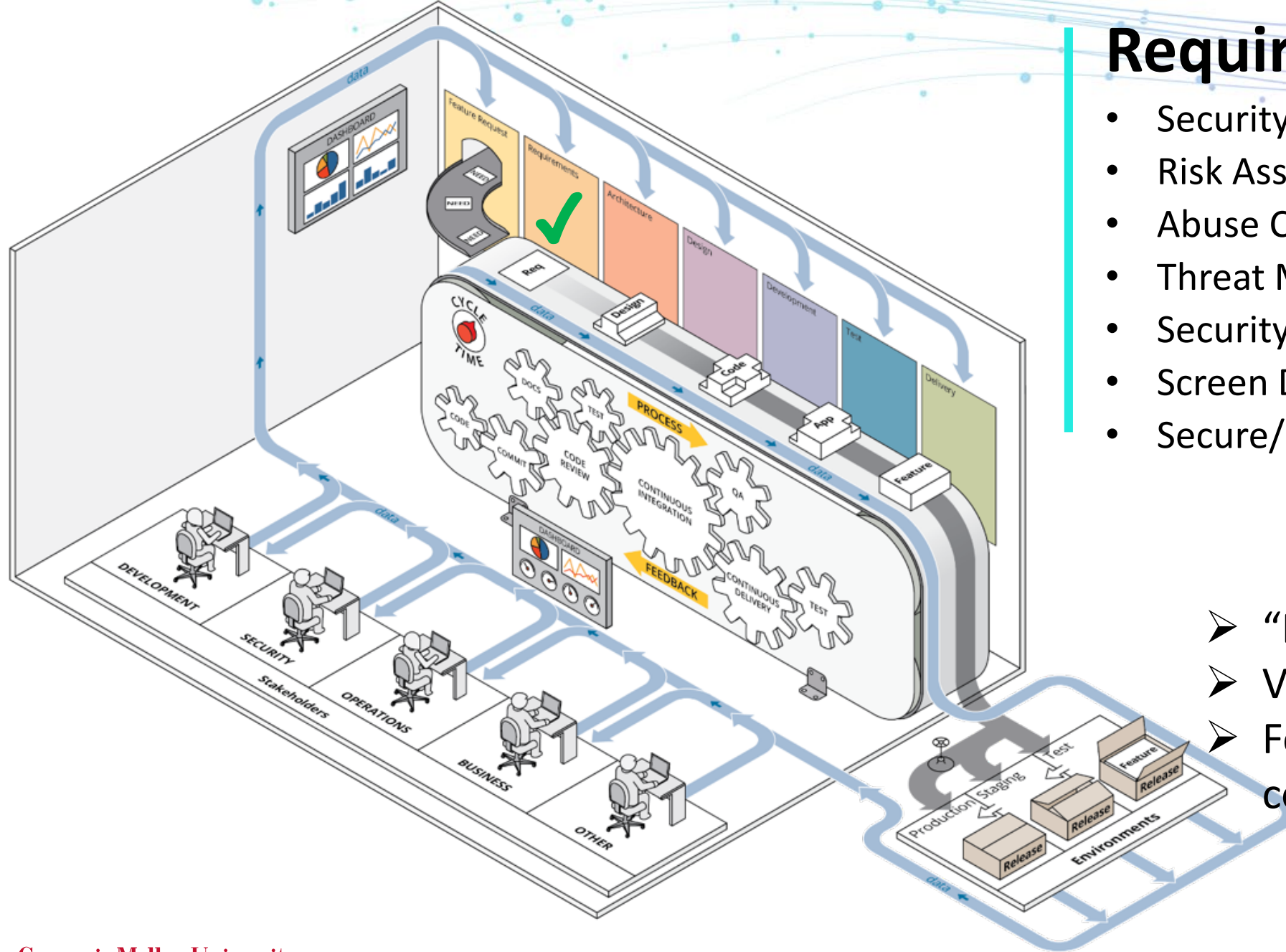


Requirements

- Security Requirements (SFR/SAR)
- Risk Assessment
- Abuse Case Development
- Threat Modelling
- Security Stories
- Screen Development Tools
- Secure/Hardened Environments

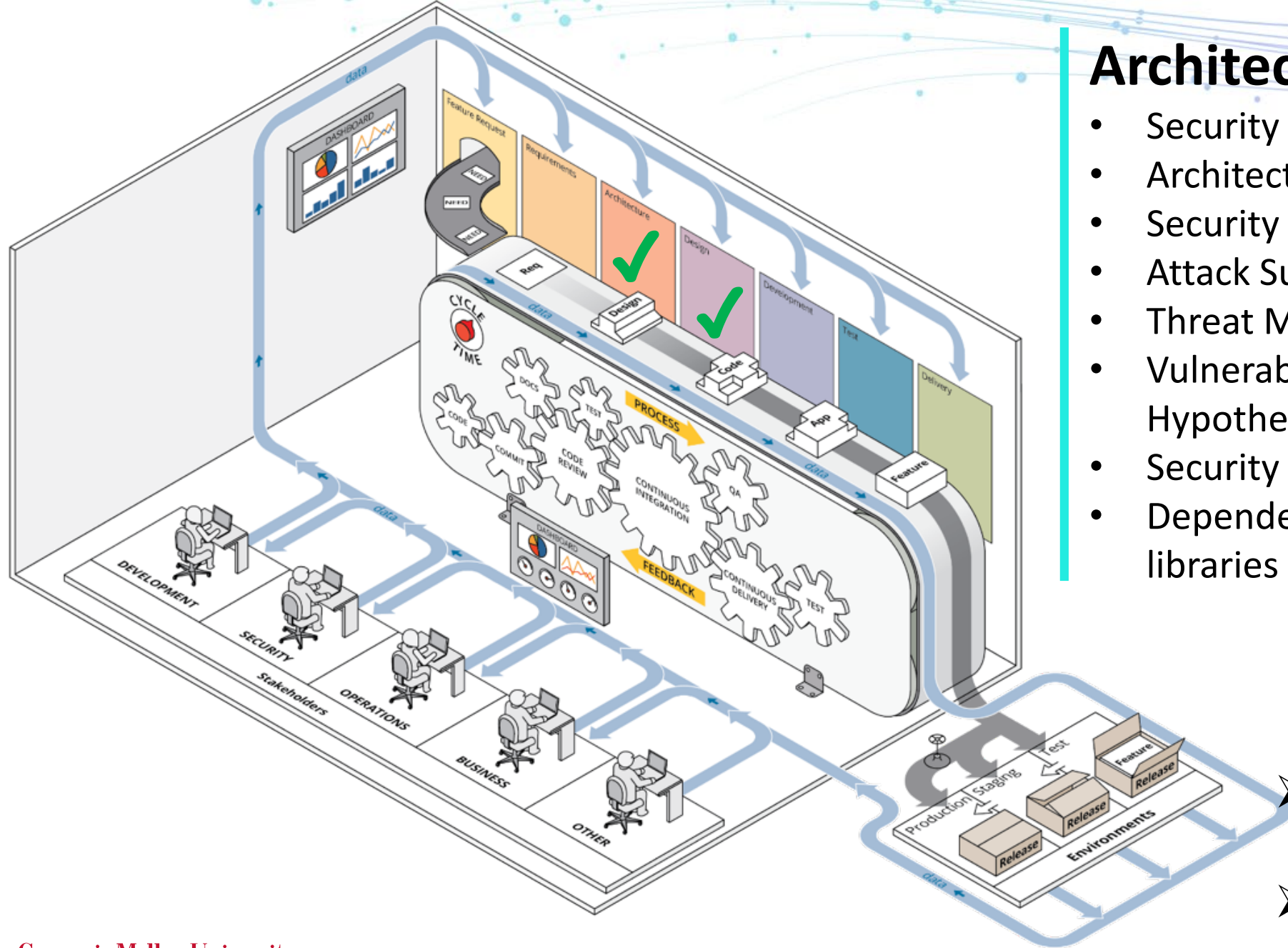


- “Baked in” Security Thoughts
- Verify Security Requirements
- Feature based security controls



Architecture & Design

- Security Architecture
- Architectural Risk Analysis
- Security Design Requirements
- Attack Surface Analysis
- Threat Modelling
- Vulnerability Analysis and Flow Hypothesis
- Security Design Review
- Dependencies List, Open-source libraries



Verify and Validate
Security Design

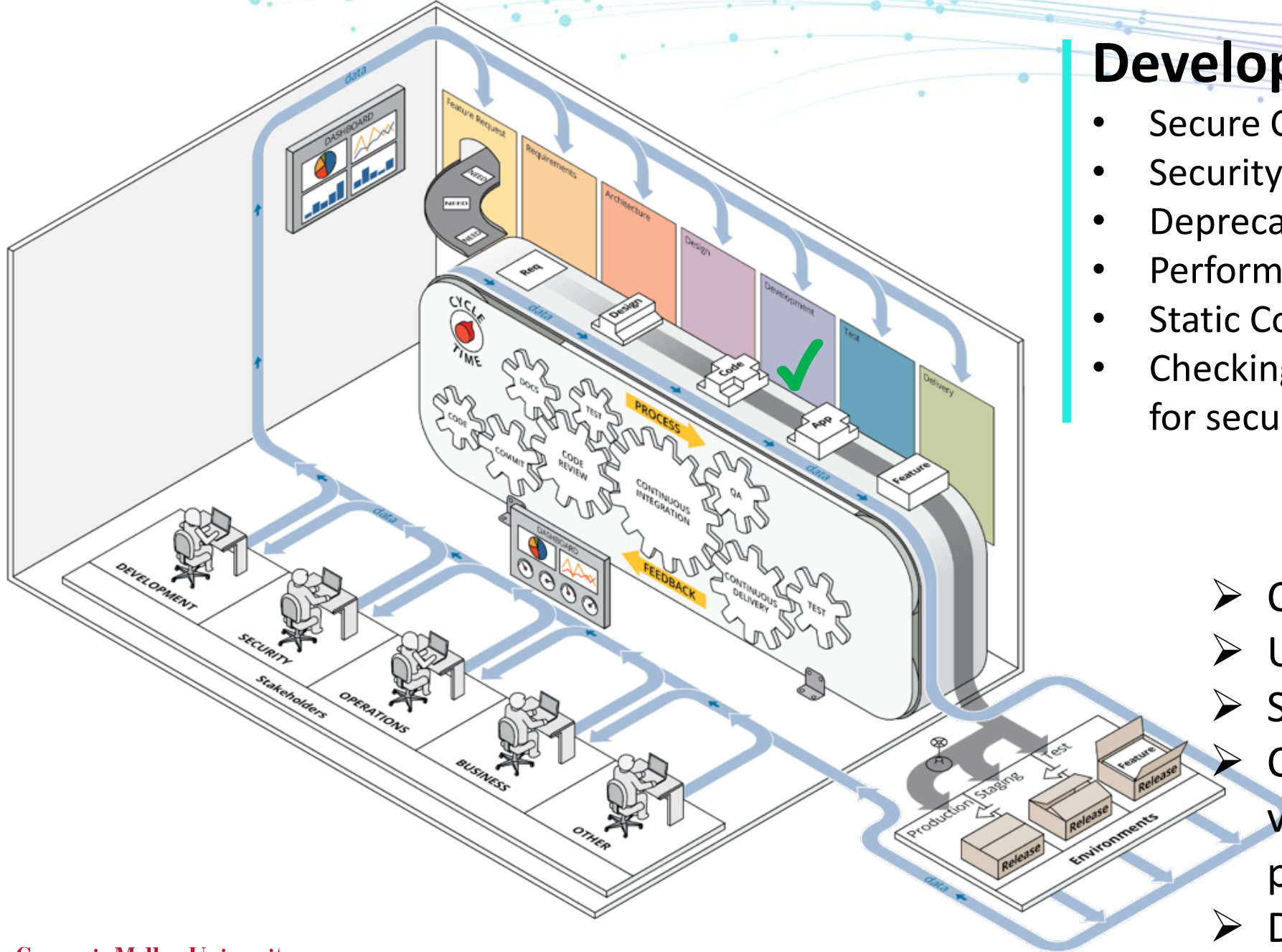
➤ Personnel data- privacy

Development

- Secure Coding Practices
- Security Focused Code Review
- Deprecate Unsafe Functions
- Perform Security Unit Testing
- Static Code Analysis
- Checking of process and procedures for secure coding & traceability



- Code Development Audit
- Unit Testing result
- Static Code Analysis results
- Code verification and validation on security practices
- Design validation

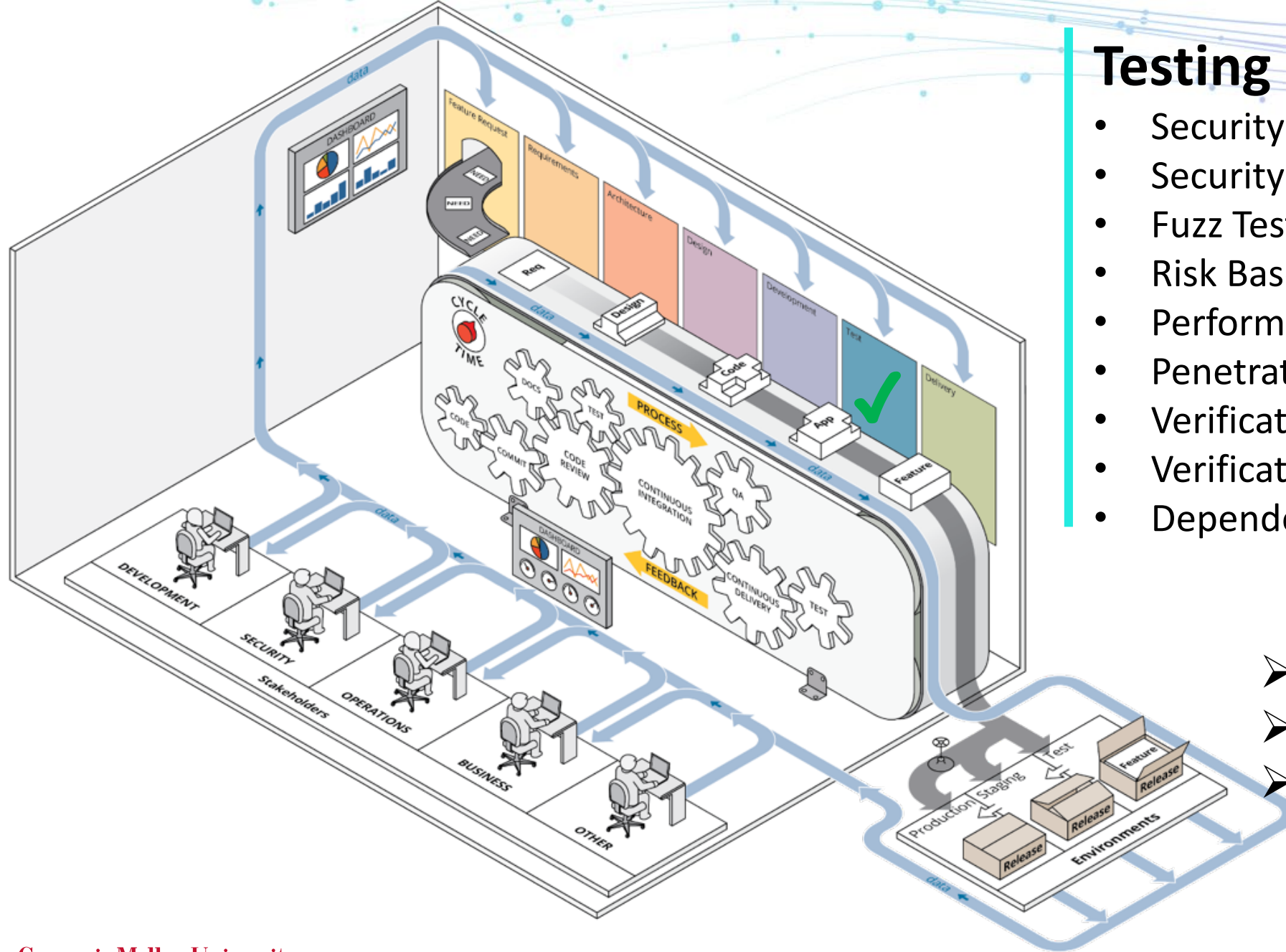


Testing

- Security Test Planning
- Security Testing
- Fuzz Testing
- Risk Based Security Testing
- Perform Dynamic Analysis
- Penetration Testing
- Verification of Security Implementation
- Verification of Process and Procedures
- Dependency Monitoring



- Test results,
- Data handling variation
- Validation of security features

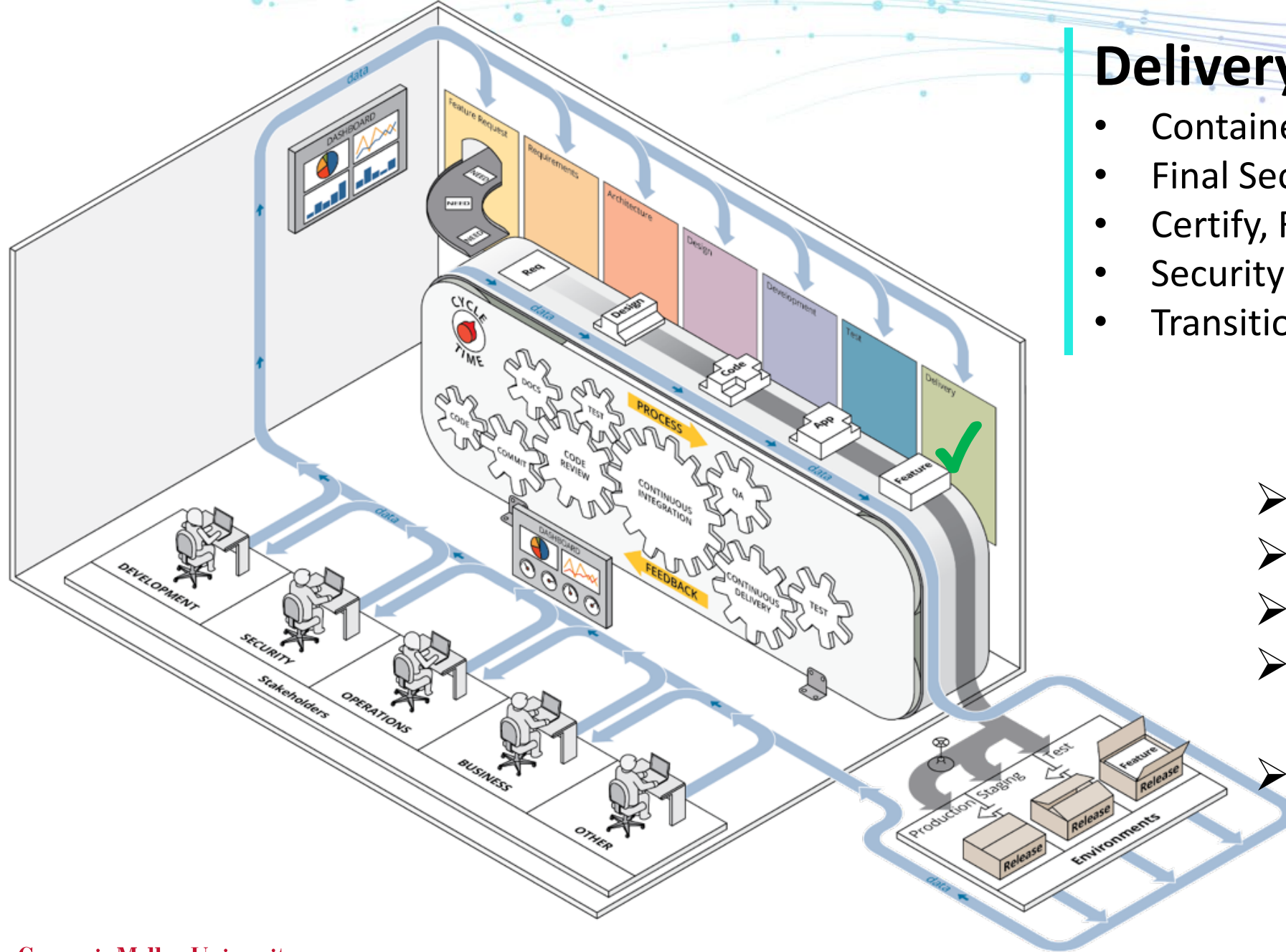


Delivery

- Container Security
- Final Security Review
- Certify, Release and Archive
- Security Acceptance Testing
- Transition Incident Response Plan



- Pre-approval
- Dependency checks
- Validate incident response
- Audit data access /rights /contents
- Environment verification

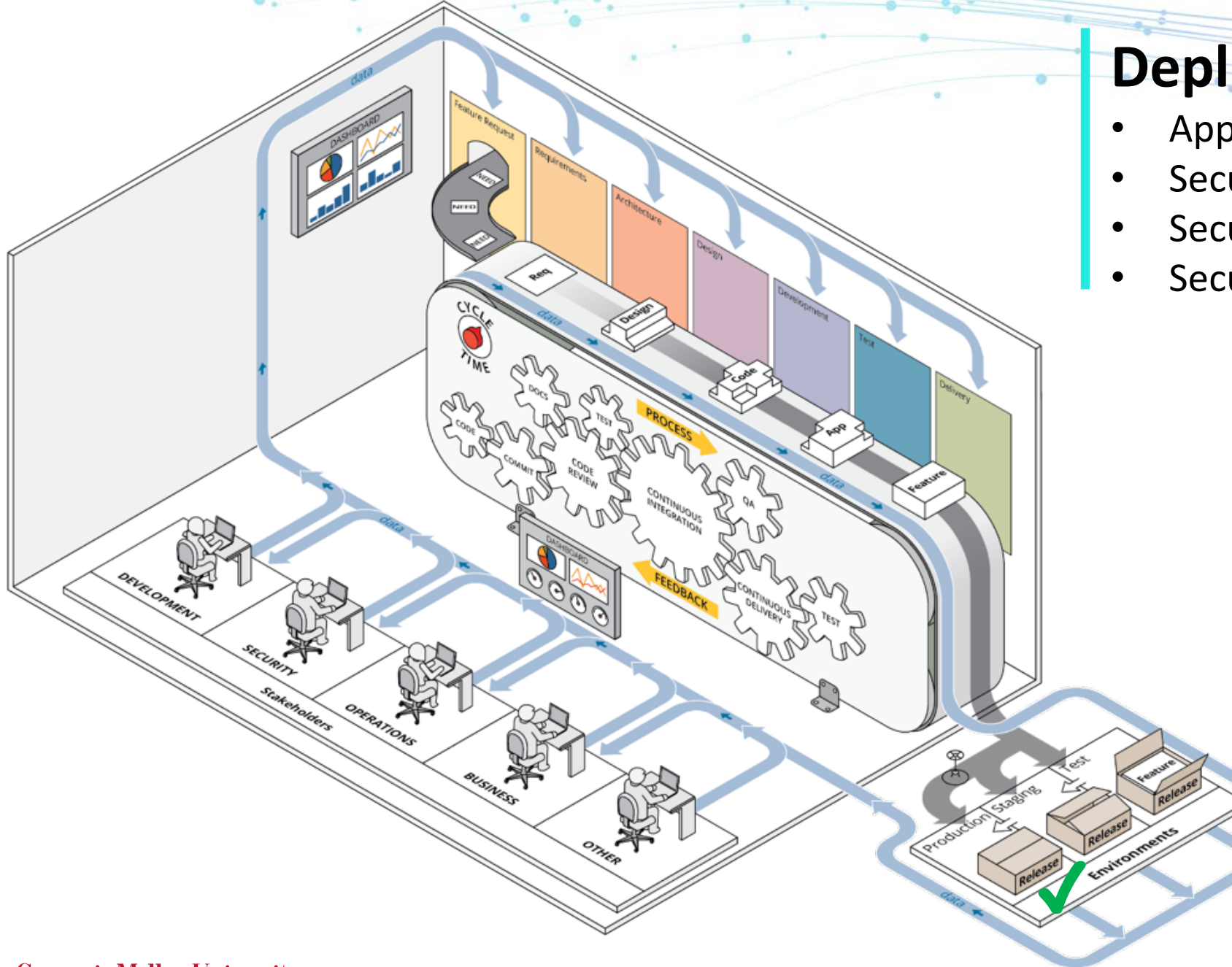


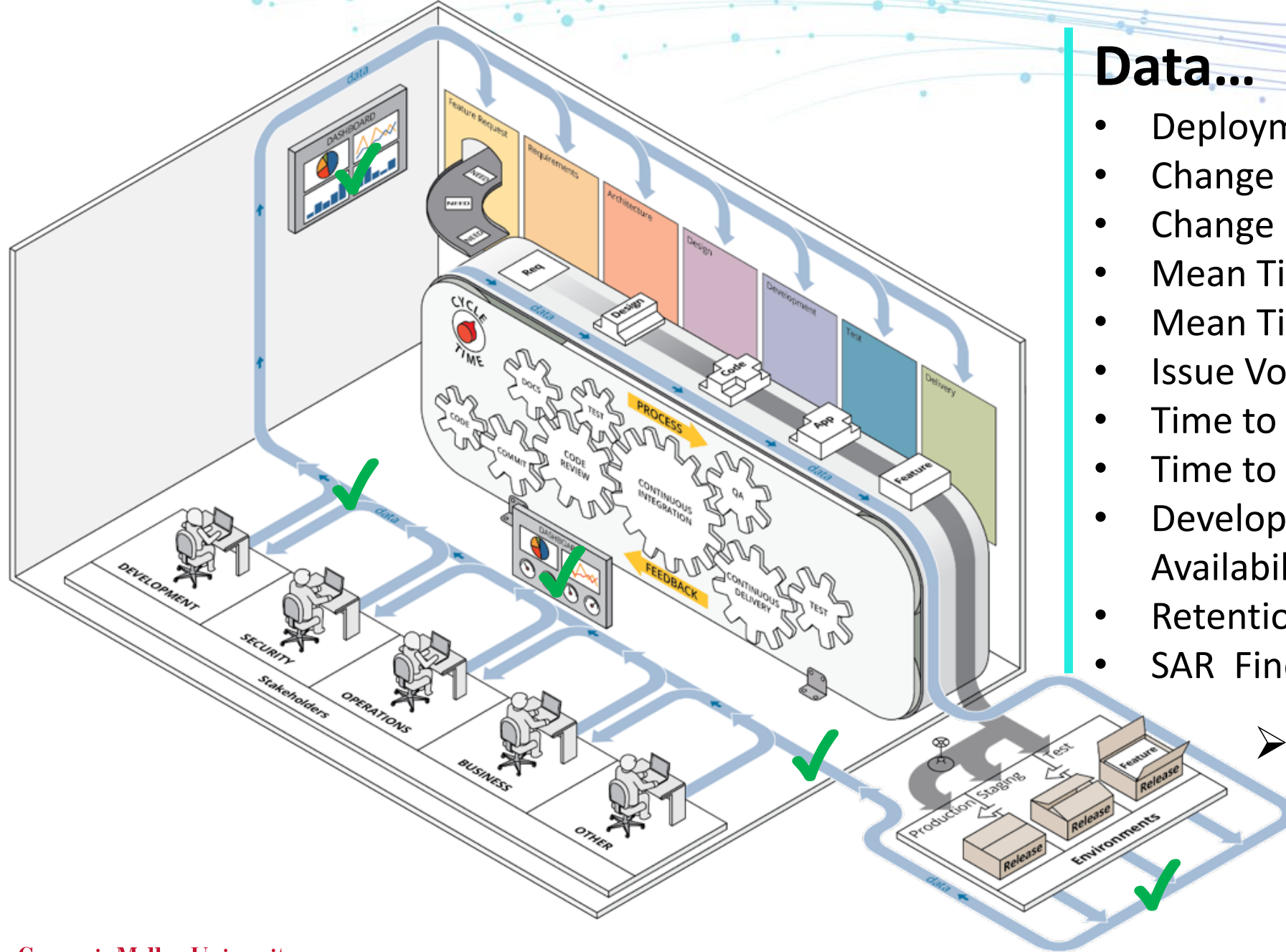
Deploy

- Application Security Monitoring
- Secure Deployment Process
- Secure Environment
- Secure Operational Enablement



- Security Dashboard
- Security Status
- Incident Response
- Rollback capabilities
- Application /Environments logs
- IDS/IPS logs
- Environment monitoring
- Resource usage
- Data handling process

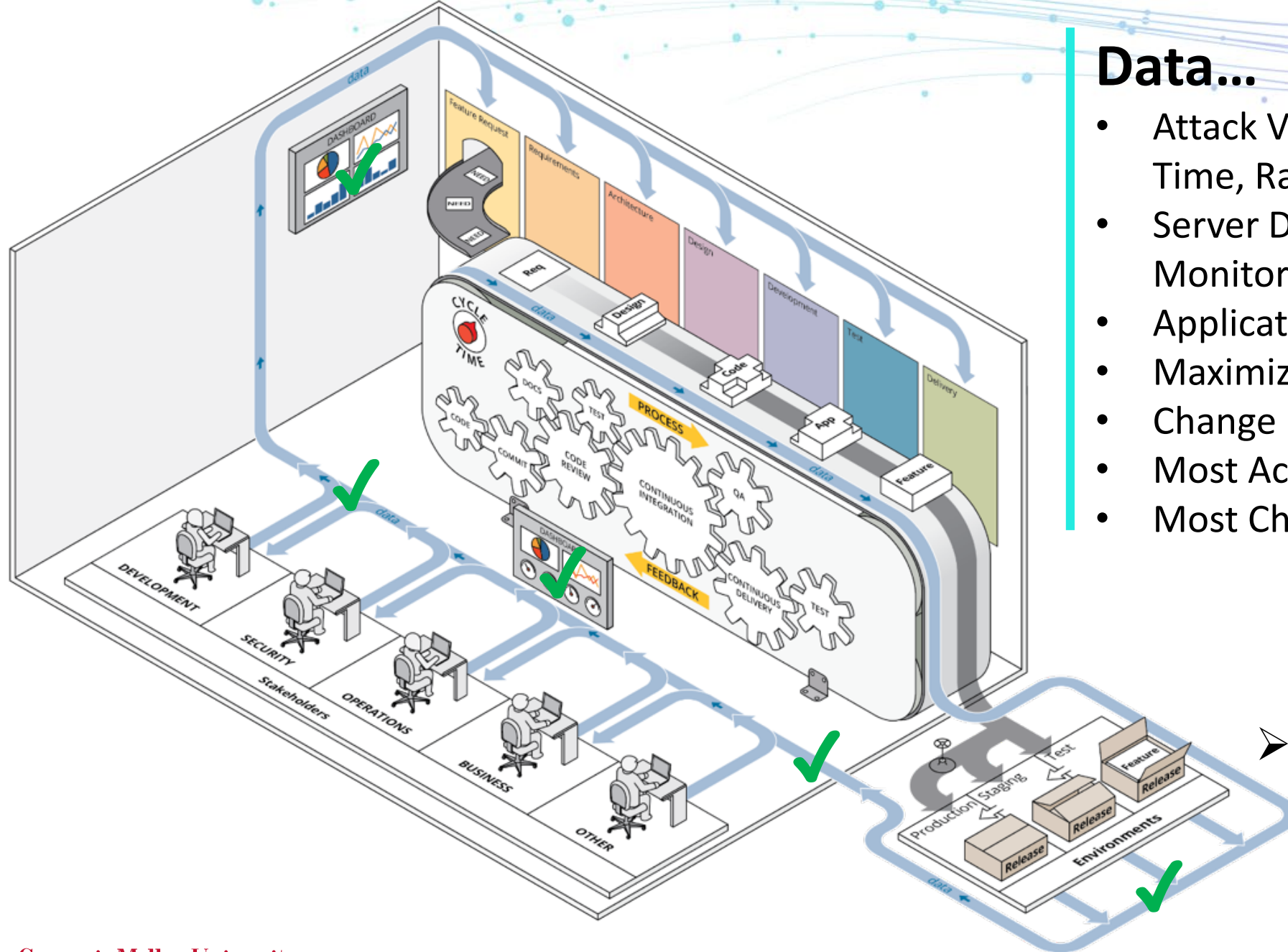




Data...

- Deployment Frequency
- Change Lead Time and Volume
- Change Failure Rate
- Mean Time To Recovery (MTTR)
- Mean Time to Detection (MTTD)
- Issue Volume and Resolution Time
- Time to Approval
- Time to Patch Vulnerabilities
- Development and Application Logging Availability
- Retention Control Compliance
- SAR Findings

➤ Continuous Monitoring to feed **Continuous Security**



Data...

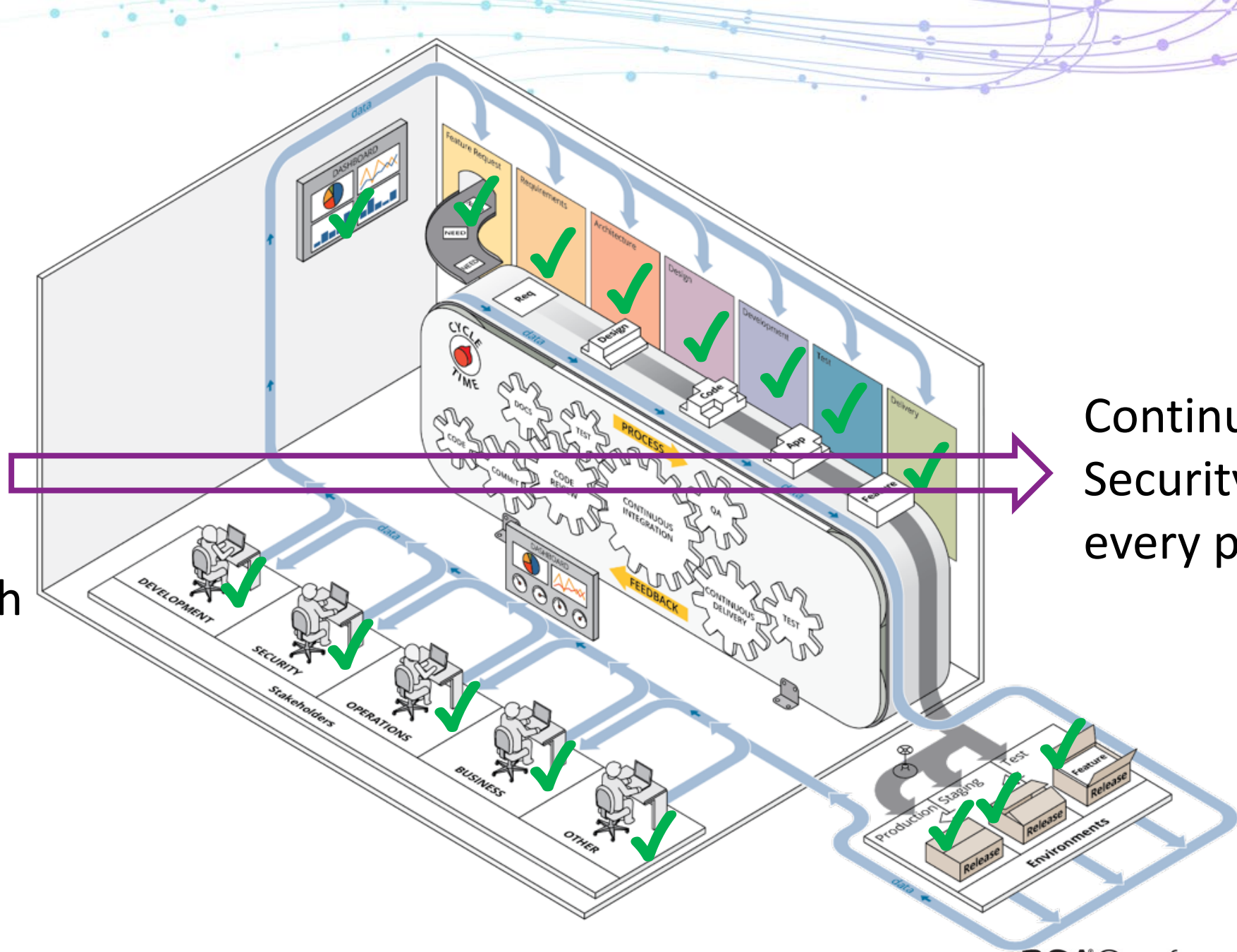
- Attack Vector Details (IP, Stack Trace, Time, Rate of Attack, etc)
- Server Disk Space, Load and Process Monitoring
- Application Performance
- Maximize Monitoring
- Change in Size to Code Base
- Most Active Code Contributors
- Most Changed Code Areas



➤ Continuous Monitoring to feed **Continuous Security**

Be Ready..

Security from inception to deployment and improvement with every delivery



Continuous Security on every phases

Poll the Audience

- ASD-W02
- Do you believe that we will have a more secure system with DevSecOps?
 - A. YES
 - B. No
- <https://rsa1-live.eventbase.com/polls?event=rsa2019&polls=3859>

Apply What You Have Learned Today

- Next week you should:
 - Start to communicate with all stakeholders
 - Inventory any automation process/scripts/code etc..
 - Understand your organization security framework/compliances
 - Build DevSecOps pipeline
- In the first three months following this presentation you should:
 - Build a transparent and visible collaboration platform
 - Analyze automation process
 - Insert Security controls as part of CI/CD
- Within six months you should:
 - Apply secure automation process across the DevSecOps pipeline
 - Build visible common dashboard
 - Automate security controls on each possible phases of system lifecycle.

For more information...

Go DevOps!

sei.cmu.edu/go/devops

DevOps Blog

insights.sei.cmu.edu/devops

Webinar

sei.cmu.edu/publications/webinars

Podcast

sei.cmu.edu/publications/podcasts

SLS team GitHub Projects

- Once Click DevOps deployment
github.com/SLS-ALL/devops-microcosm
- Sample app with DevOps Process
[github.com/SLS-ALL/flask api sample](https://github.com/SLS-ALL/flask_api_sample)
- Tagged checkpoints
 - v0.1.0: base Flask project
 - v0.2.0: Vagrant development configuration
 - v0.3.0: Test environment and Fabric deployment
 - v0.4.0: Upstart services, external configuration files
 - v0.5.0: Production environment
- On YouTube:
<https://youtu.be/5nQIJ-FWA5A>

Any Questions?

Hasan Yasar

**Technical Manager,
Secure Lifecycle Solutions**

hyasar@sei.cmu.edu

@securelifecycle

