GitLab

Securing The Software Supply Chain

The importance of DevSecOps Governance and Value Stream Management in the Era of AI-Powered Software Development.

GitLab

Tomasz Skora

Senior Solutions Architect

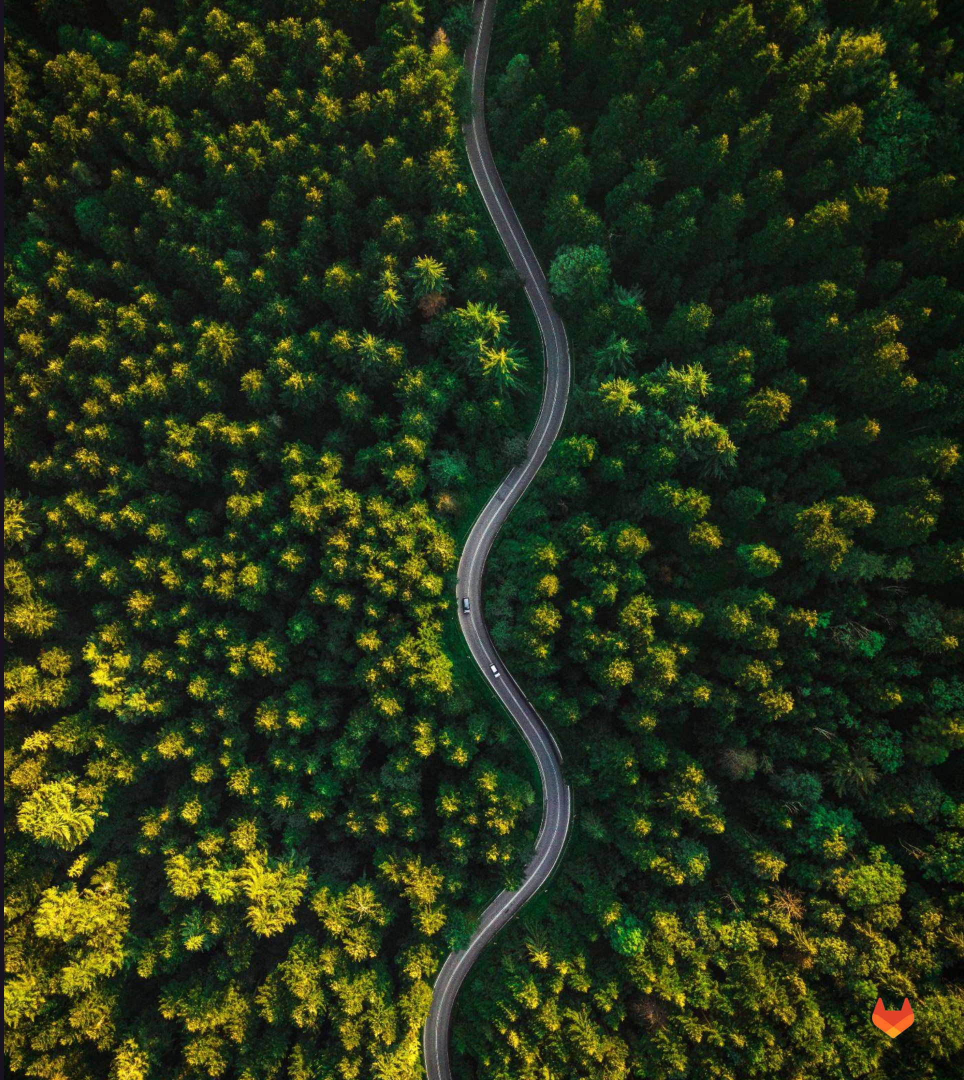
GitLab

Connect with me on LinkedIn in

Agenda

1. Introduction
2. Value Stream Management (VSM)
3. AI Impact on DevSecOps
4. Securing the Speed
5. DevSecOps & AI & Value Stream Management

Introduction



Our Findings

In March 2023, we asked more than 5,000 DevSecOps professionals to share their opinions on the current state of software development, operations, and security.

65%

of developers said they are using artificial intelligence and machine learning in testing efforts or will be in the next three years.

2023 Gitlab DevSecOps Survey

Our Findings

How can we secure the Software Supply Chain considering security budget constraints?

62%

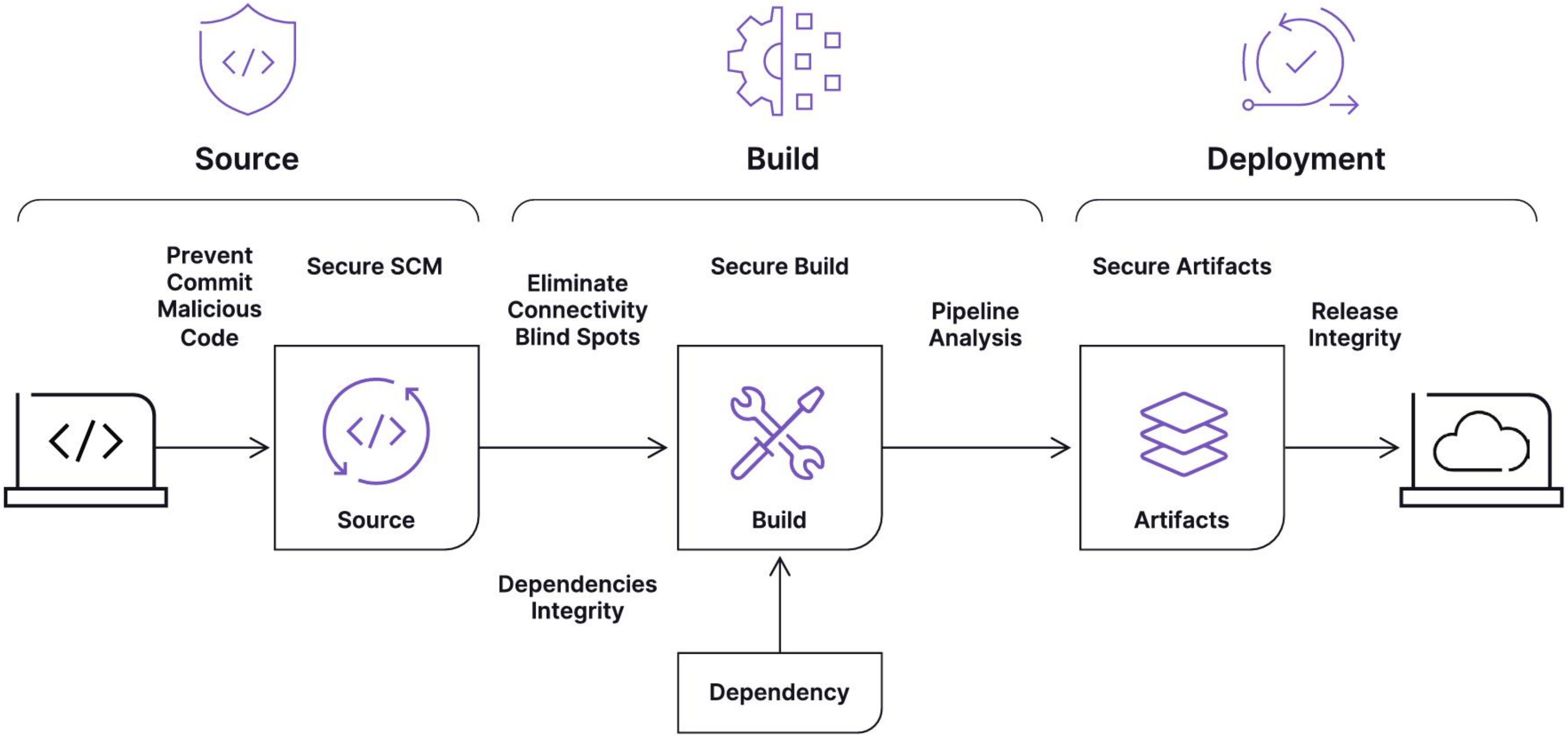
have shifted left or are planning to shift left this year.

85%

of the security professionals have the same or less budget this year than they did in 2022.

2023 Gitlab DevSecOps Survey

Software Supply Chain Attacks



Value Stream Management

Key Questions

How to **identify** security inefficiencies and bottlenecks in the business value delivery and DevSecOps pipeline?

How to **measure** the flow of value from ideation to production, focusing on DevSecOps effectiveness?

How to **visualise** and **optimise** end-to-end DevSecOps workstream?

What is a Value Stream?

A value stream is an end-to-end set of activities which collectively creates value for the customer.

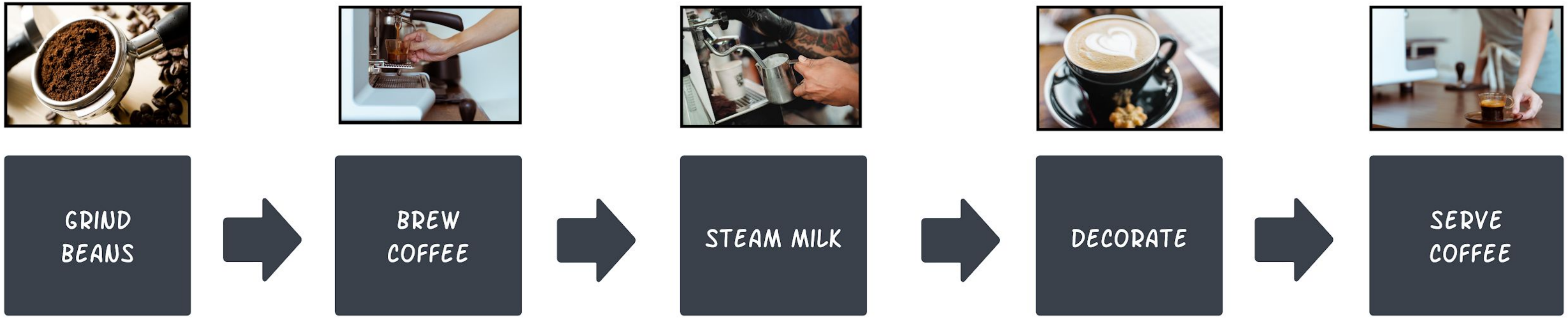
Source: book *"The Great Transition", James Martin*

GitLab

Cafe Shop

Value Stream Cappuccino Example

Cafe Shop Value Stream Example



GRIND BEANS

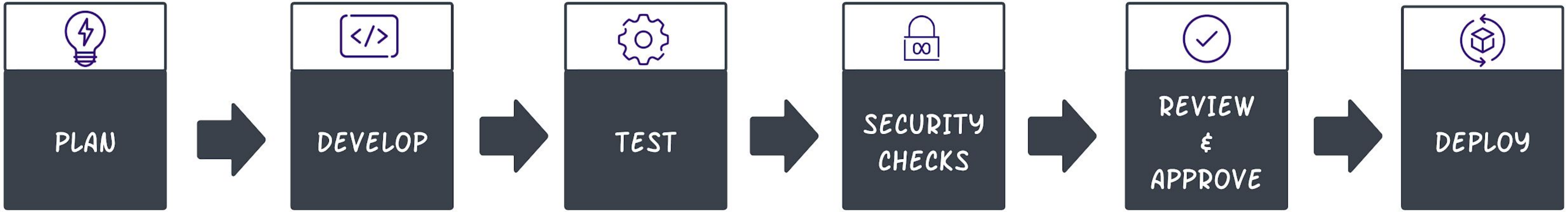
BREW COFFEE

STEAM MILK

DECORATE

SERVE COFFEE

DevSecOps Value Stream Example



PLAN

DEVELOP

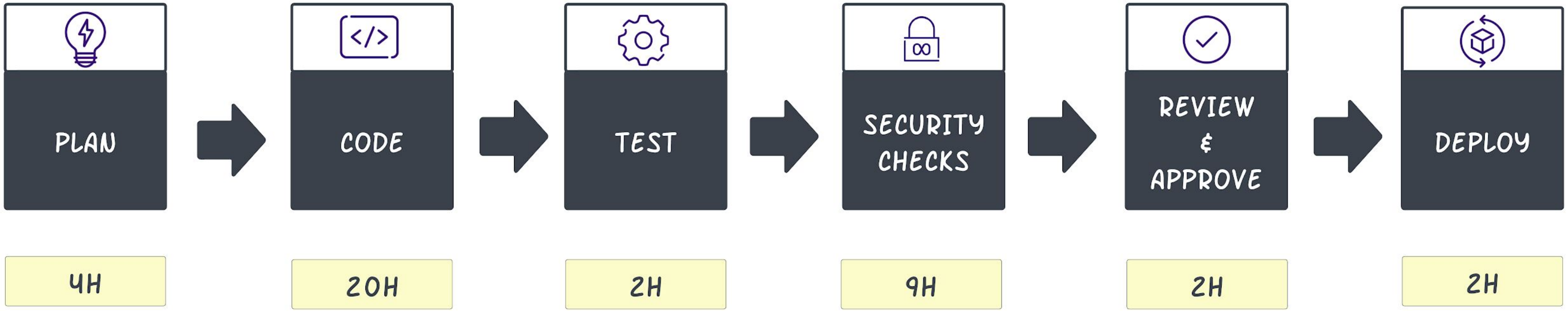
TEST

SECURITY CHECKS

REVIEW & APPROVE

DEPLOY

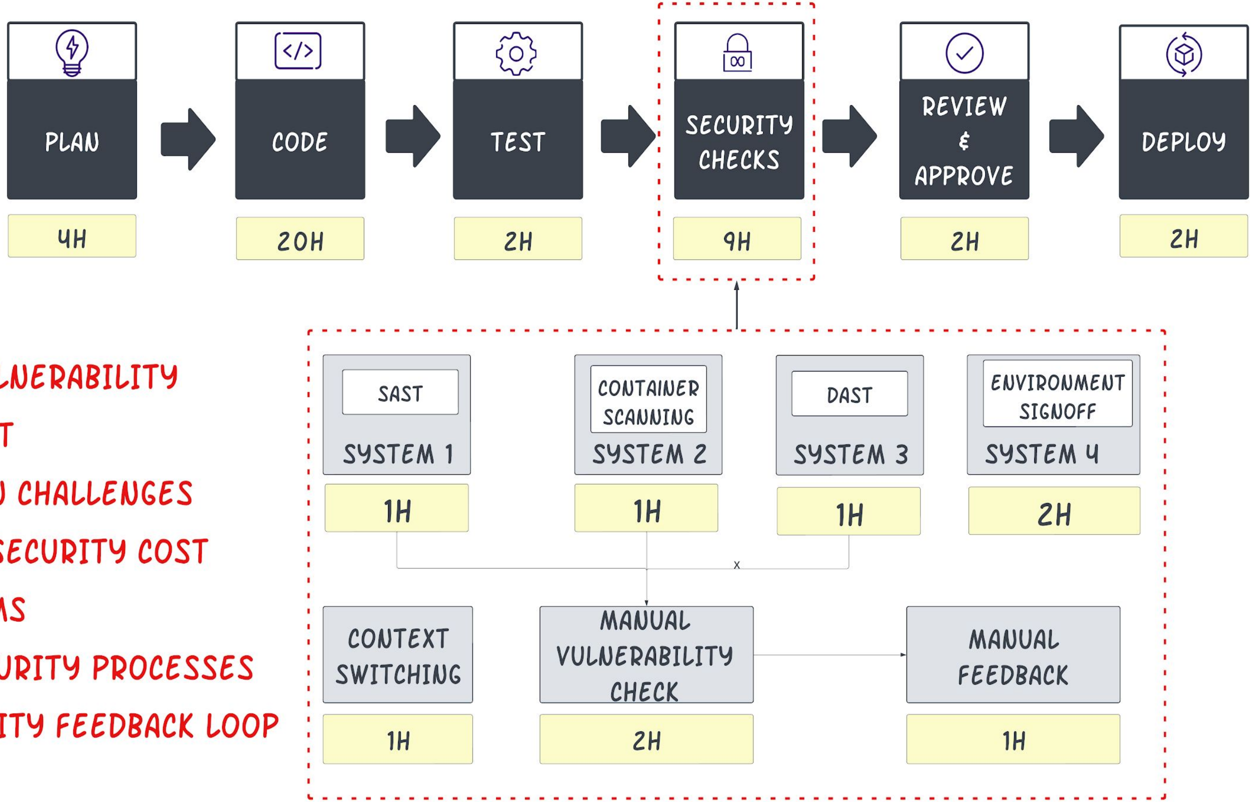
DevSecOps Value Stream Example



PLAN → CODE → TEST → SECURITY CHECKS → REVIEW & APPROVE → DEPLOY

PLAN	CODE	TEST	SECURITY CHECKS	REVIEW & APPROVE	DEPLOY
4H	20H	2H	9H	2H	2H

DevSecOps Value Stream Example - Security Inefficiencies



PLAN

4H

CODE

20H

TEST

2H

SECURITY CHECKS

9H

REVIEW & APPROVE

2H

DEPLOY

2H

SAST

SYSTEM 1

1H

CONTAINER SCANNING

SYSTEM 2

1H

DAST

SYSTEM 3

1H

ENVIRONMENT SIGNOFF

SYSTEM 4

2H

CONTEXT SWITCHING

1H

MANUAL VULNERABILITY CHECK

2H

MANUAL FEEDBACK

1H

- COMPLEX VULNERABILITY MANAGEMENT
- INTEGRATION CHALLENGES
- INCREASED SECURITY COST
- SILOED TEAMS
- MANUAL SECURITY PROCESSES
- SLOW SECURITY FEEDBACK LOOP

“If you can’t describe what you are doing as a value stream, you don’t know what you’re doing.”

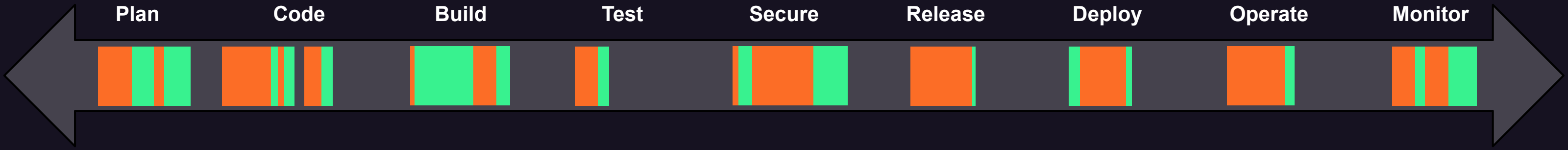
Source: book *Martin, K. & Osterling, M. (2014). Value Stream Mapping. McGraw-Hill*

What is your current and future DevSecOps state?

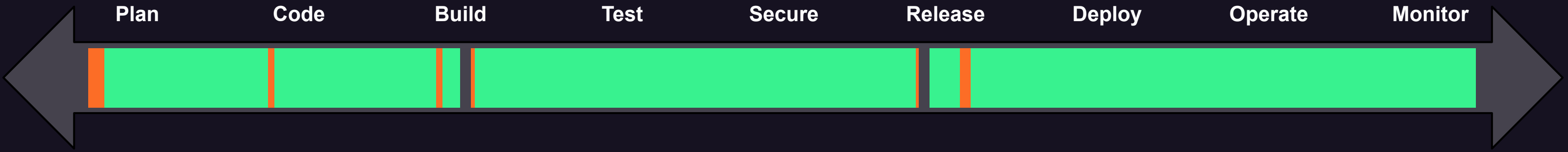
and how you measure your progress

- ● Value Added Time
- ● Non-Value Added Time
- ● Idle Time

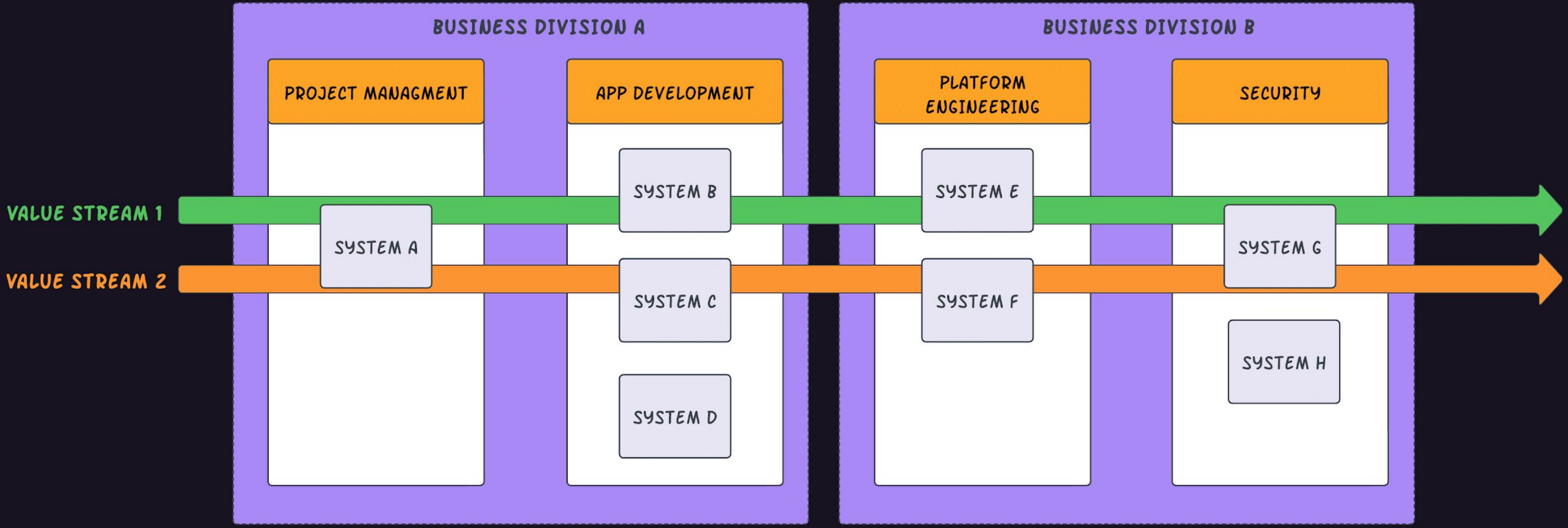
Current DevSecOps State



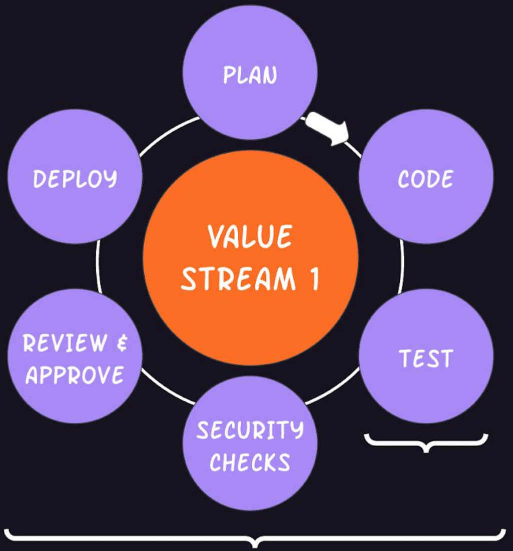
Desired DevSecOps Future State



Value Streams can be hard to measure ...

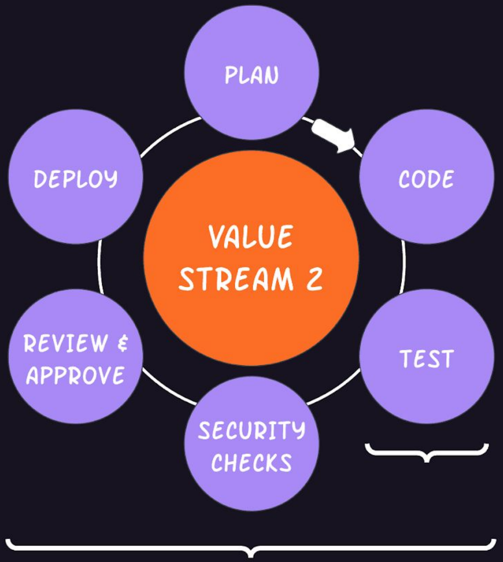


Multiple Value Streams & Security Management



SECURITY VISIBILITY?
SECURITY BOTTLENECKS?
DEVSECOPS EFFECTIVENES?

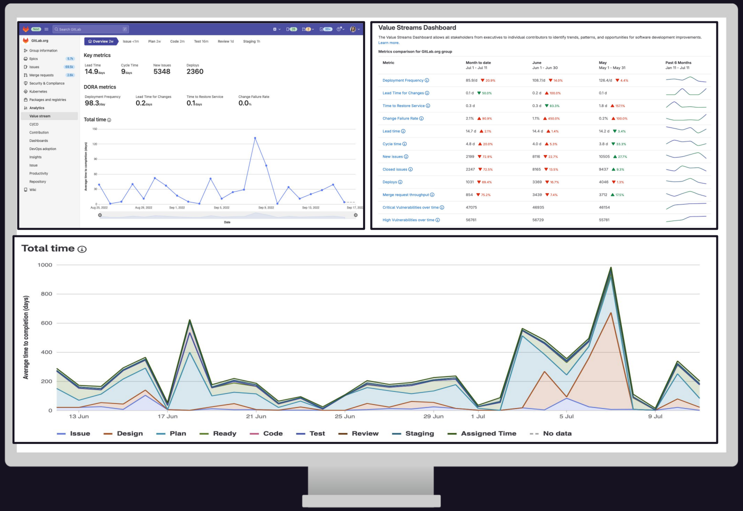
CYBER SECURITY PROGRAM
REQUIRED BUDGET
CYBER SECURITY REPORTING
SYSTEM SECURITY STATUS
CYBER SECURITY STRATEGY



SECURITY VISIBILITY?
SECURITY BOTTLENECKS?
DEVSECOPS EFFECTIVENES?

GitLab Value Stream Management (VSM) enables executive visibility across value streams.

- ✓ **Value streams dashboards** and metrics to identify security bottlenecks and deficiencies resulting in improved visibility into the organization's security posture.
- ✓ **Holistic visibility** and platform approach allows allows security leaders to gain a comprehensive understanding of security performance, facilitating informed decision-making.
- ✓ **Improved collaboration** to align security goals with other teams, fostering a shared understanding of security objectives.



Key metrics

Lead Time 14.9 days · Cycle Time 9 days · New Issues 5348 · Deploys 2360

DORA metrics

Deployment Frequency 98.3 /day · Lead Time for Changes 0.2 days · Time to Restore Service 0.1 days · Change Failure Rate 0.0 %

Total time

Value Streams Dashboard

The Value Streams Dashboard allows all stakeholders from executives to individual contributors to identify trends, patterns, and opportunities for software development improvements. Learn more.

Metrics comparison for GitLab.org group

Metric	Month to date Jul 1 - Jul 11	June Jun 1 - Jun 30	May May 1 - May 31	Past 6 Months Jan 11 - Jul 11
Deployment Frequency	85.9/d ▼ 20.9%	108.7/d ▼ 14.0%	126.4/d ▼ 4.4%	
Lead Time for Changes	0.1 d ▼ 50.0%	0.2 d ▲ 100.0%	0.1 d	
Time to Restore Service	0.3 d	0.3 d ▼ 83.3%	1.8 d ▲ 157.1%	
Change Failure Rate	2.1% ▲ 90.9%	1.1% ▲ 450.0%	0.2% ▲ 100.0%	
Lead time	14.7 d ▲ 2.1%	14.4 d ▲ 1.4%	14.2 d ▼ 3.4%	
Cycle time	4.8 d ▲ 20.0%	4.0 d ▲ 5.3%	3.8 d ▼ 33.3%	
New issues	2199 ▼ 72.9%	8116 ▼ 22.7%	10505 ▲ 27.7%	
Closed issues	2247 ▼ 72.5%	8165 ▼ 13.5%	9437 ▲ 9.3%	
Deploys	1031 ▼ 69.4%	3369 ▼ 16.7%	4046 ▼ 1.3%	
Merge request throughput	854 ▼ 75.2%	3439 ▼ 7.4%	3712 ▲ 17.5%	
Critical Vulnerabilities over time	47075	46935	46154	
High Vulnerabilities over time	56761	56729	55781	

Total time

GitLab Copyright

AI Impact on DevSecOps

Key Software Delivery Performance DORA Metrics

1. Deployment frequency
2. Lead time for changes
3. Mean time to restore
4. Change failure rate

Deployment Frequency

How often does your organisation deploy code to production?

70%

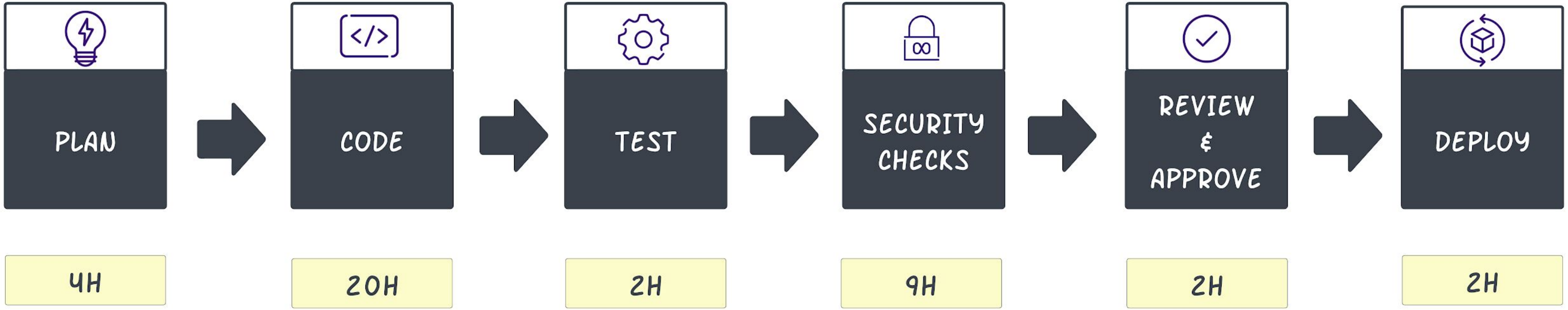
deploy at least once every few days, up 11% from 2021.

27%

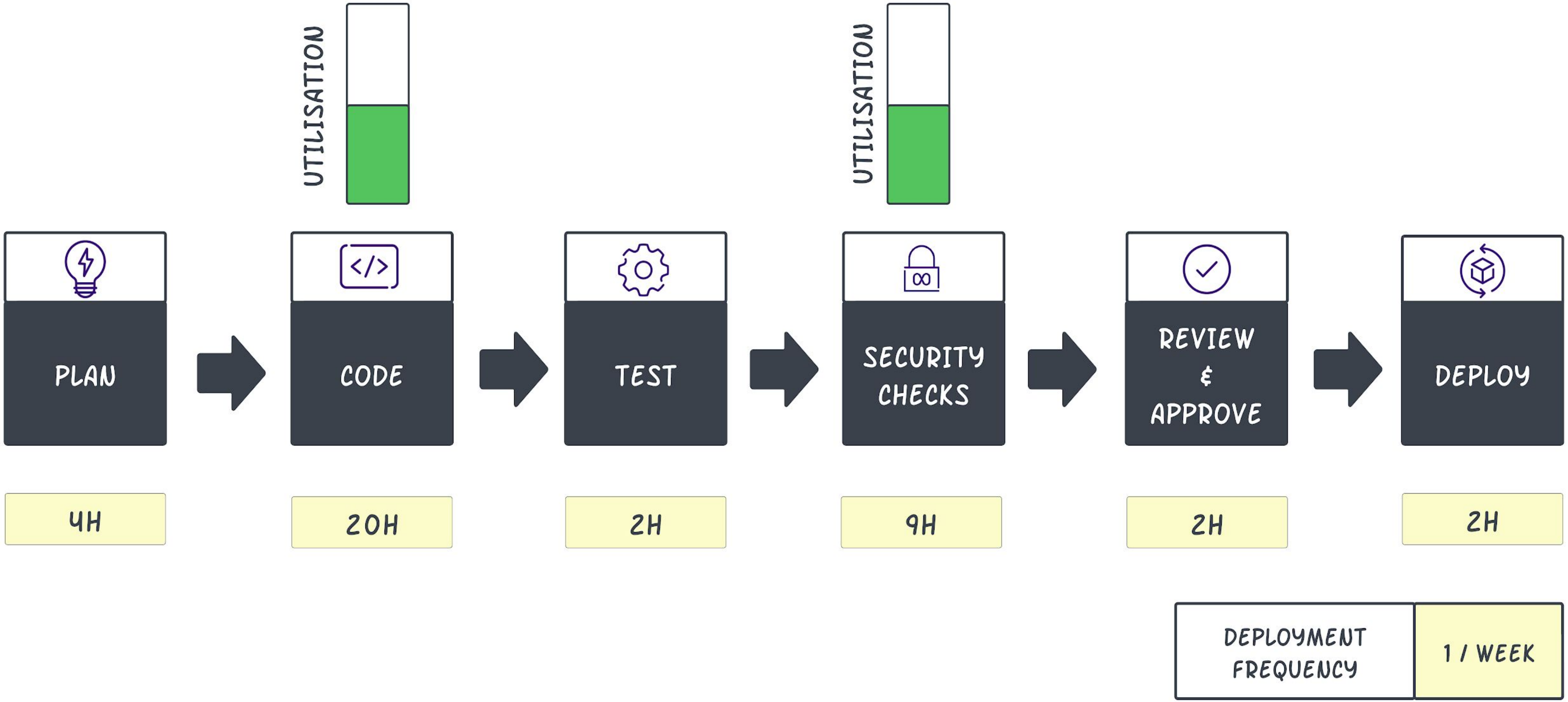
deploy multiple times a day.

2022 Gitlab DevSecOps Survey

DevSecOps Value Stream Example



DevSecOps - Value Stream Example



UTILISATION

UTILISATION

PLAN

CODE

TEST

SECURITY CHECKS

REVIEW & APPROVE

DEPLOY

4H

20H

2H

9H

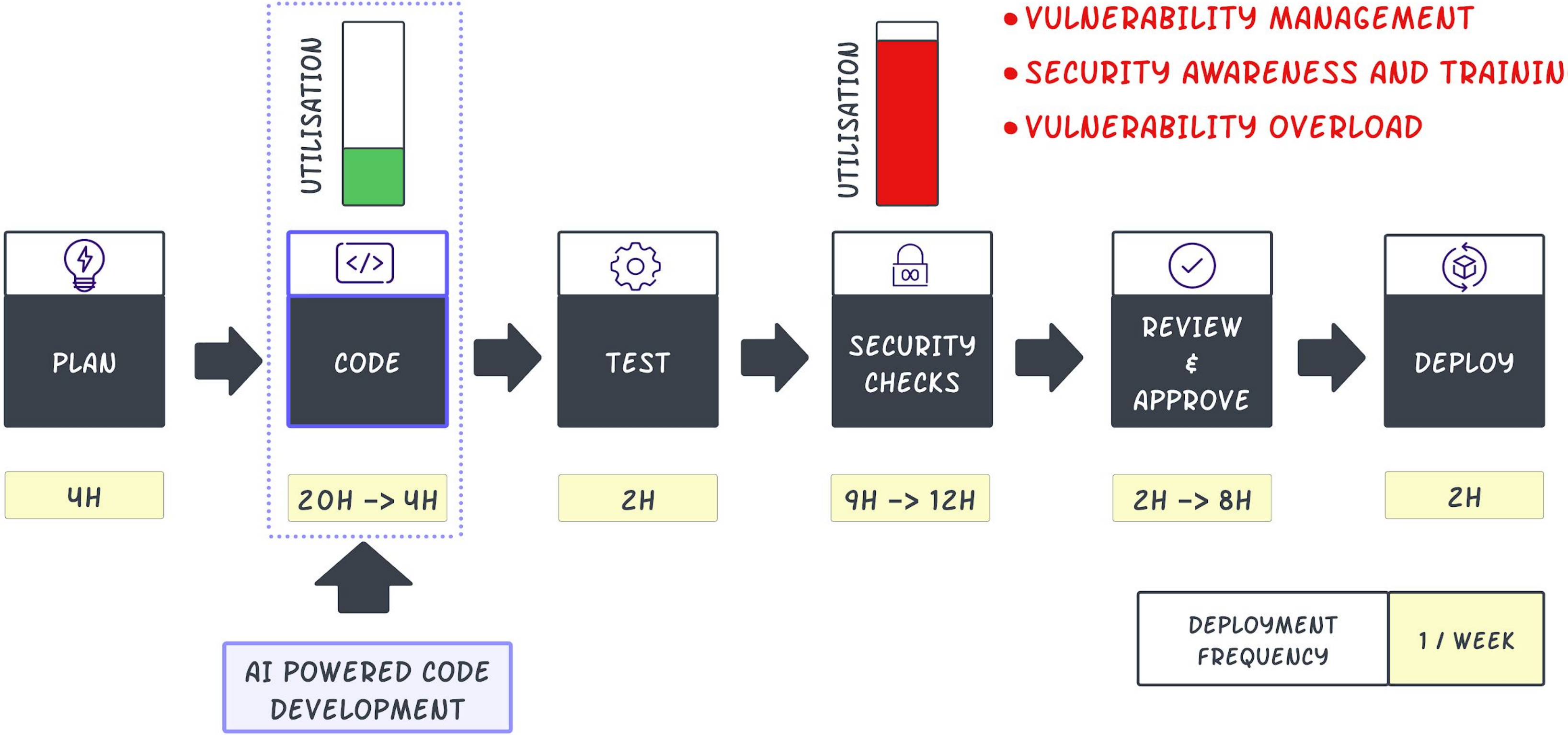
2H

2H

DEPLOYMENT FREQUENCY

1 / WEEK

Inefficient DevSecOps Value Stream Example



- SECURITY VALIDATION AND REVIEW
- VULNERABILITY MANAGEMENT
- SECURITY AWARENESS AND TRAINING
- VULNERABILITY OVERLOAD

Value Stream Management

1. Visualize DevSecOps workstreams
2. Identify risk through DevSecOps inefficiencies
3. Take action to optimize DevSecOps workstreams to deliver the highest possible velocity of value

Identify

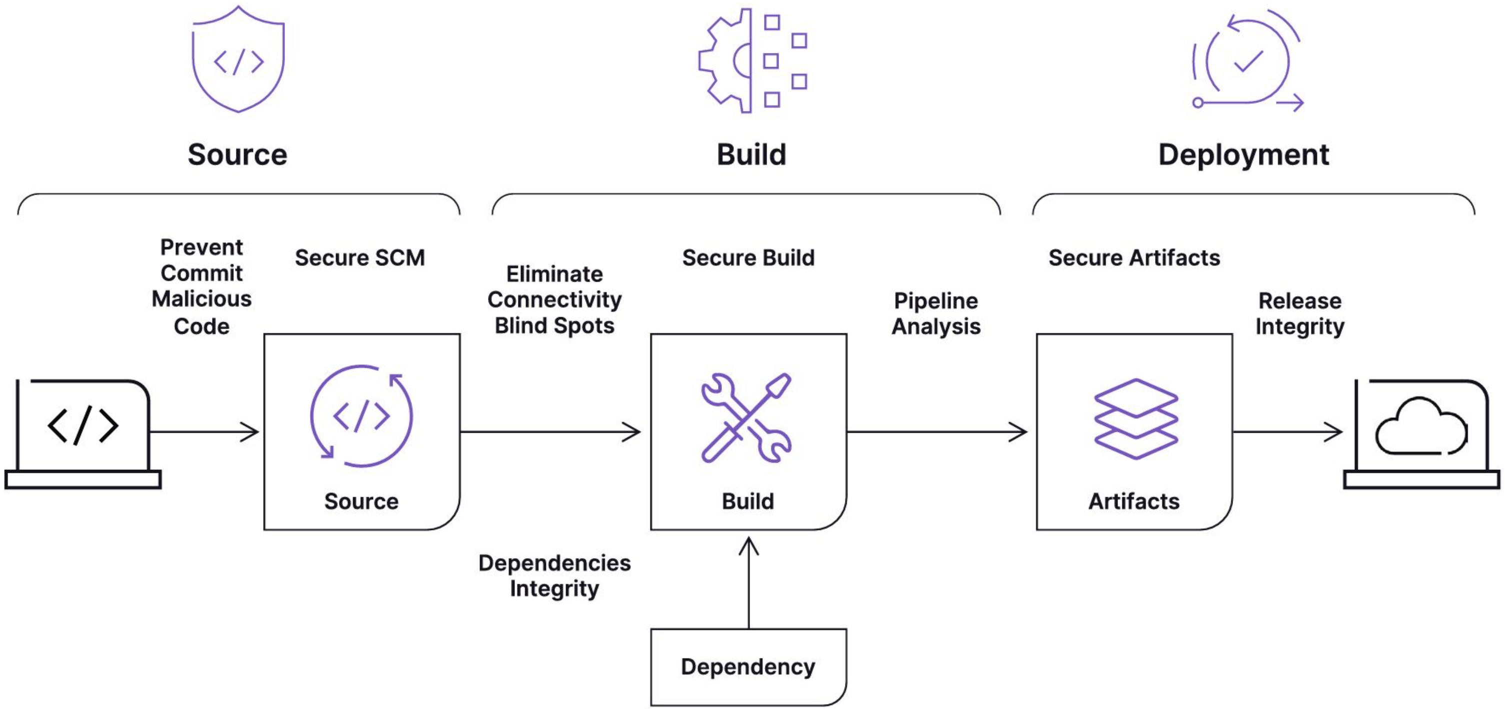
Measure

Visualise

Optimise

Securing The Speed

Software Supply Chain Security Complexity



Source

Build

Deployment

Prevent Commit Malicious Code

Secure SCM

Eliminate Connectivity Blind Spots

Secure Build

Pipeline Analysis

Secure Artifacts

Release Integrity

Source

Build

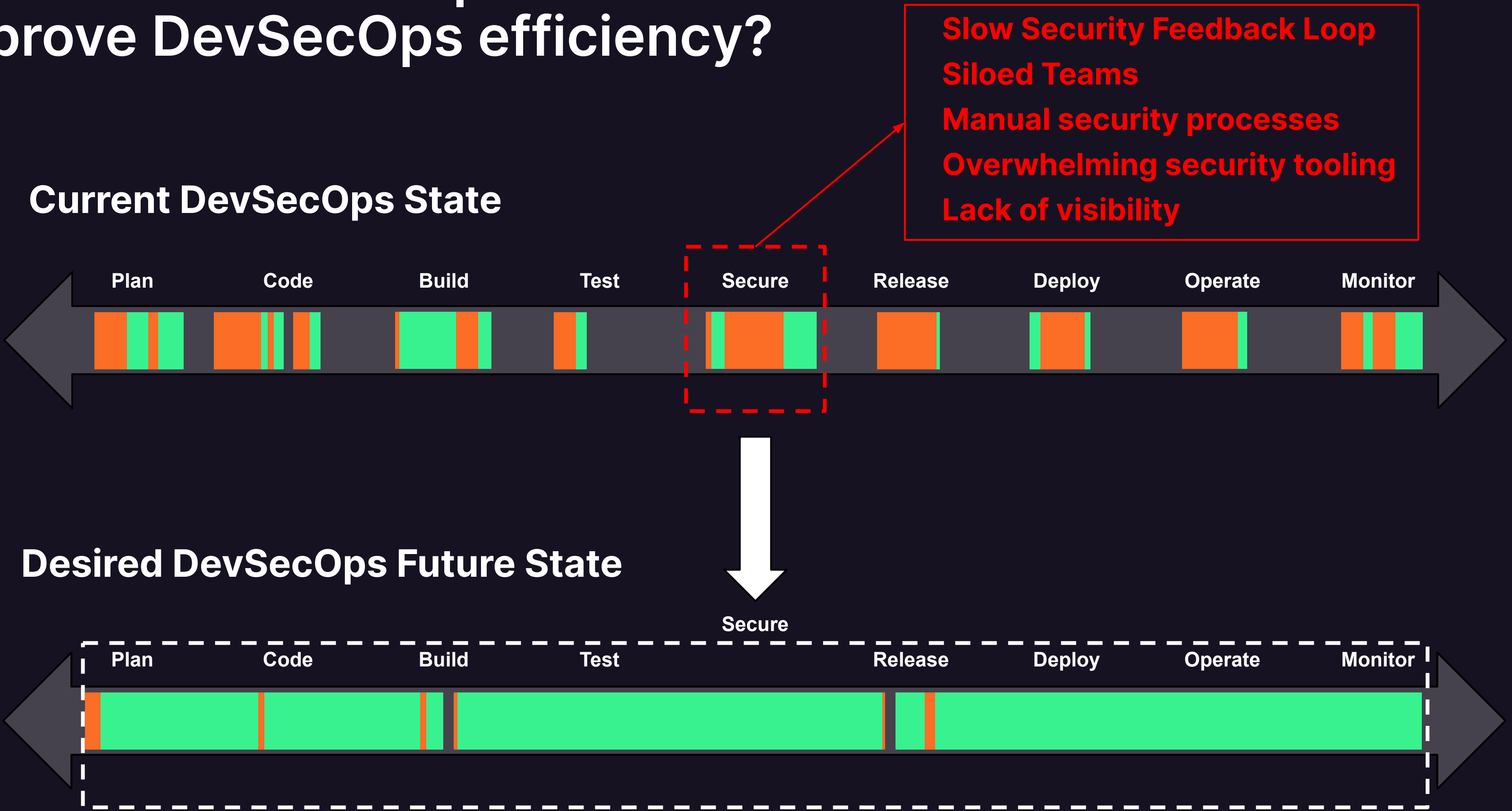
Artifacts

Dependencies Integrity

Dependency

GitLab

How to secure the speed and improve DevSecOps efficiency?



GitLab Copyright

Other Findings

1,300 CISOs were surveyed in 2022, including 100 respondents from Australia, in large organisations.

77%

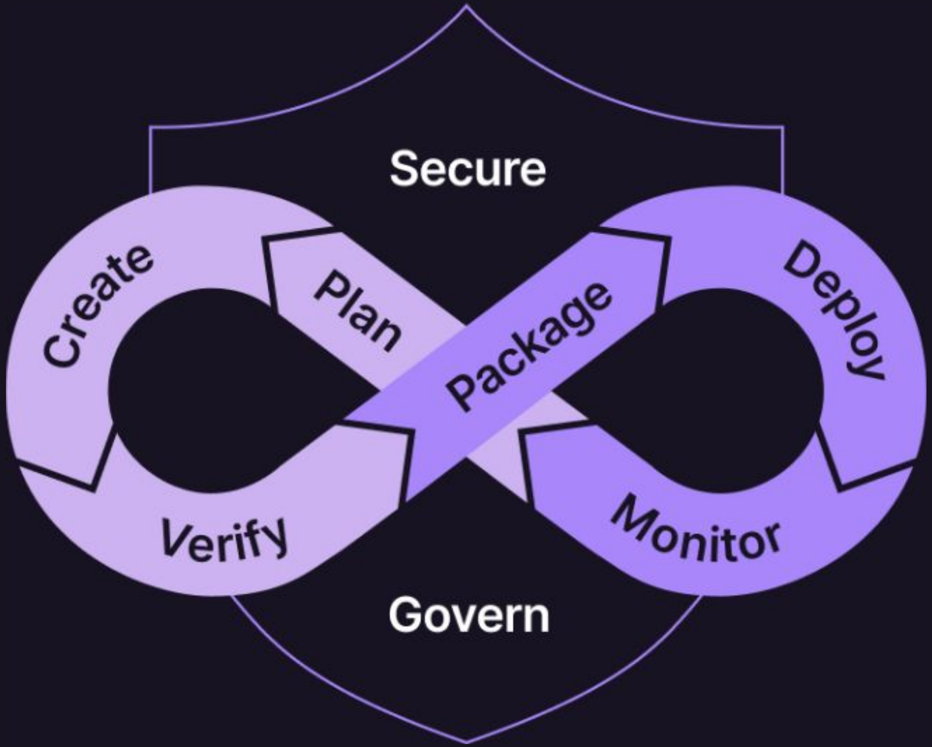
CISOs responded that the prevalence of team siloes and point solutions throughout the DevSecOps lifecycle made it easier for vulnerabilities to slip into production.

2022 Dynatrace CISO Report

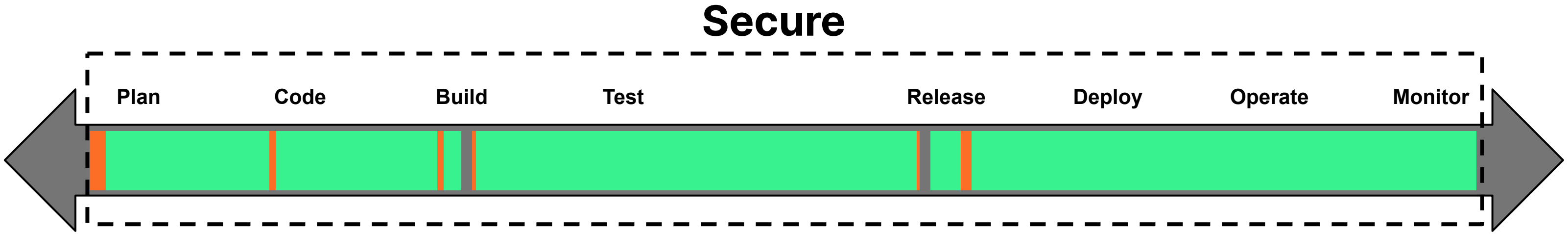
GitLab is the most comprehensive enterprise DevSecOps platform

Empower development, security, and operations teams to build better software, faster

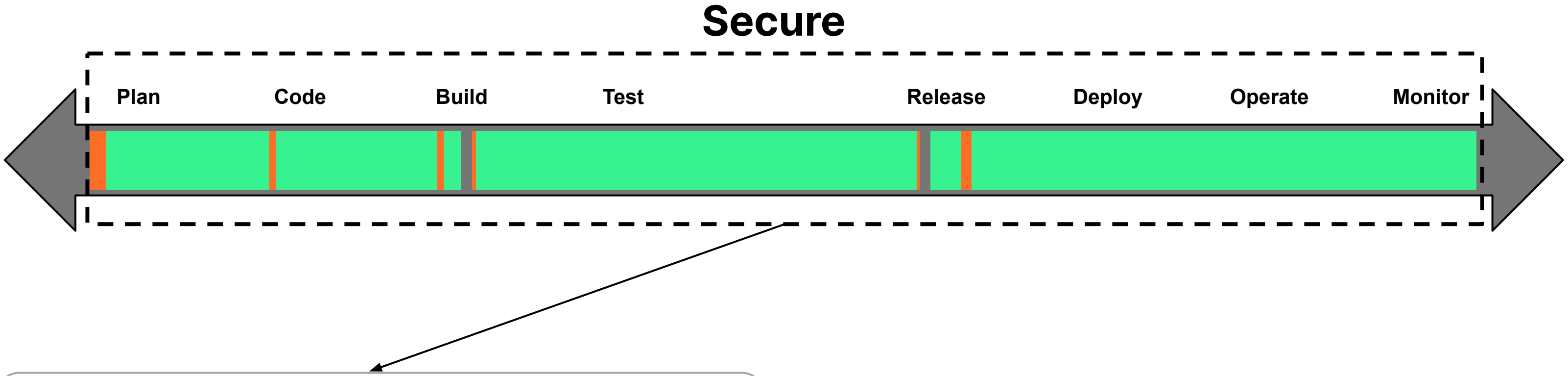
- ✓ **Better insights**: End-to-end visibility across the software delivery lifecycle
- ✓ **Greater efficiency:** built-in support for automation and integrations with third-party services
- ✓ **Improved collaboration**: One workflow that unites developer, security, and ops teams
- ✓ **Faster time to value**: Continuous improvement through accelerated feedback loops
- ✓ **Ease of Adoption**: Integrates with your existing toolchain; adopt at your own pace



Securing The Speed - DevSecOps & VSM & AI

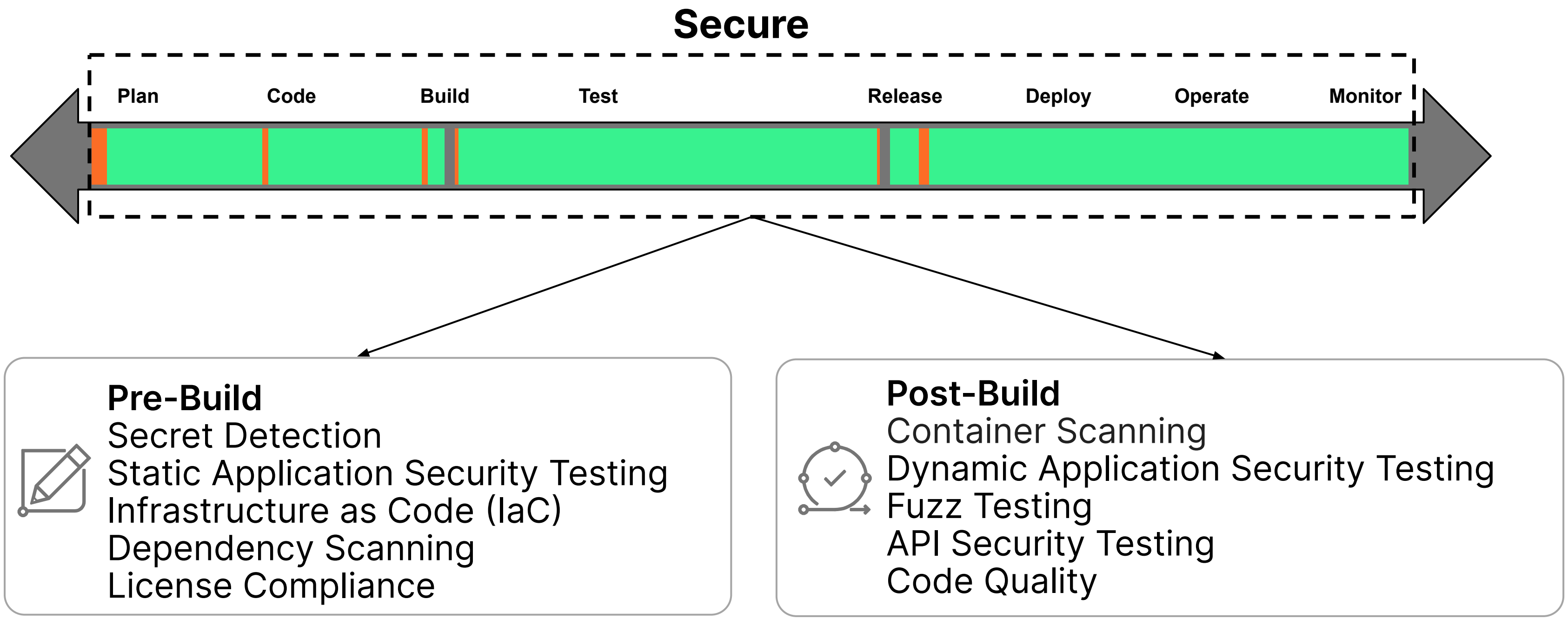


Securing The Speed - DevSecOps & VSM & AI



Pre-Build
Secret Detection
Static Application Security Testing
Infrastructure as Code (IaC)
Dependency Scanning
License Compliance

Securing The Speed - DevSecOps & VSM & AI



Pre-Build
Secret Detection
Static Application Security Testing
Infrastructure as Code (IaC)
Dependency Scanning
License Compliance

Post-Build
Container Scanning
Dynamic Application Security Testing
Fuzz Testing
API Security Testing
Code Quality

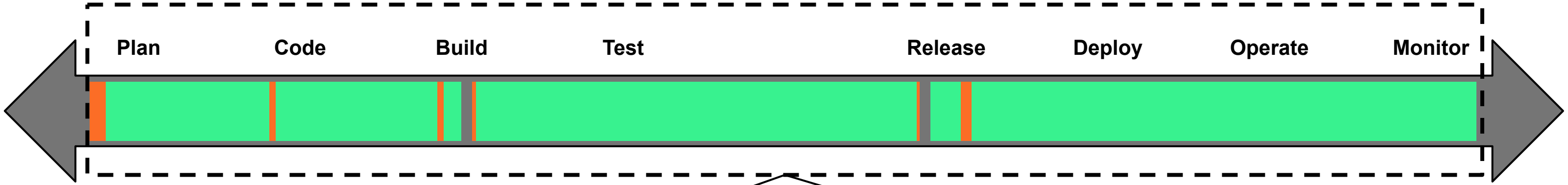
Securing The Speed - DevSecOps & VSM & AI

Governance and Scan Enforcement

Vulnerability Management

Software Bill of Materials (SBOM)

Secure



Pre-Build
Secret Detection
Static Application Security Testing
Infrastructure as Code (IaC)
Dependency Scanning
License Compliance

Post-Build
Container Scanning
Dynamic Application Security Testing
Fuzz Testing
API Security Testing
Code Quality

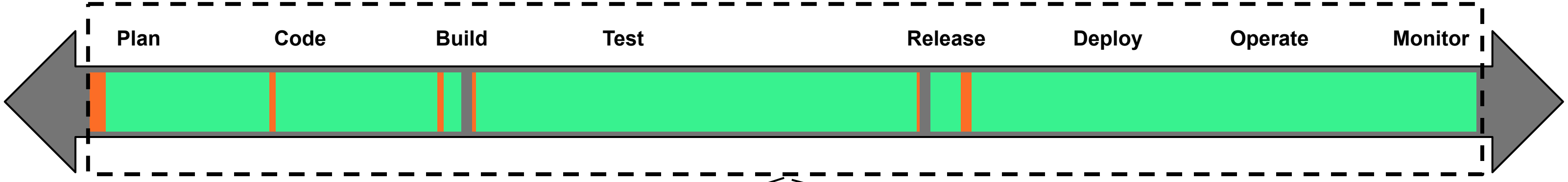
Securing The Speed - DevSecOps & VSM & AI

Governance and Scan Enforcement

Vulnerability Management

Software Bill of Materials (SBOM)

Secure

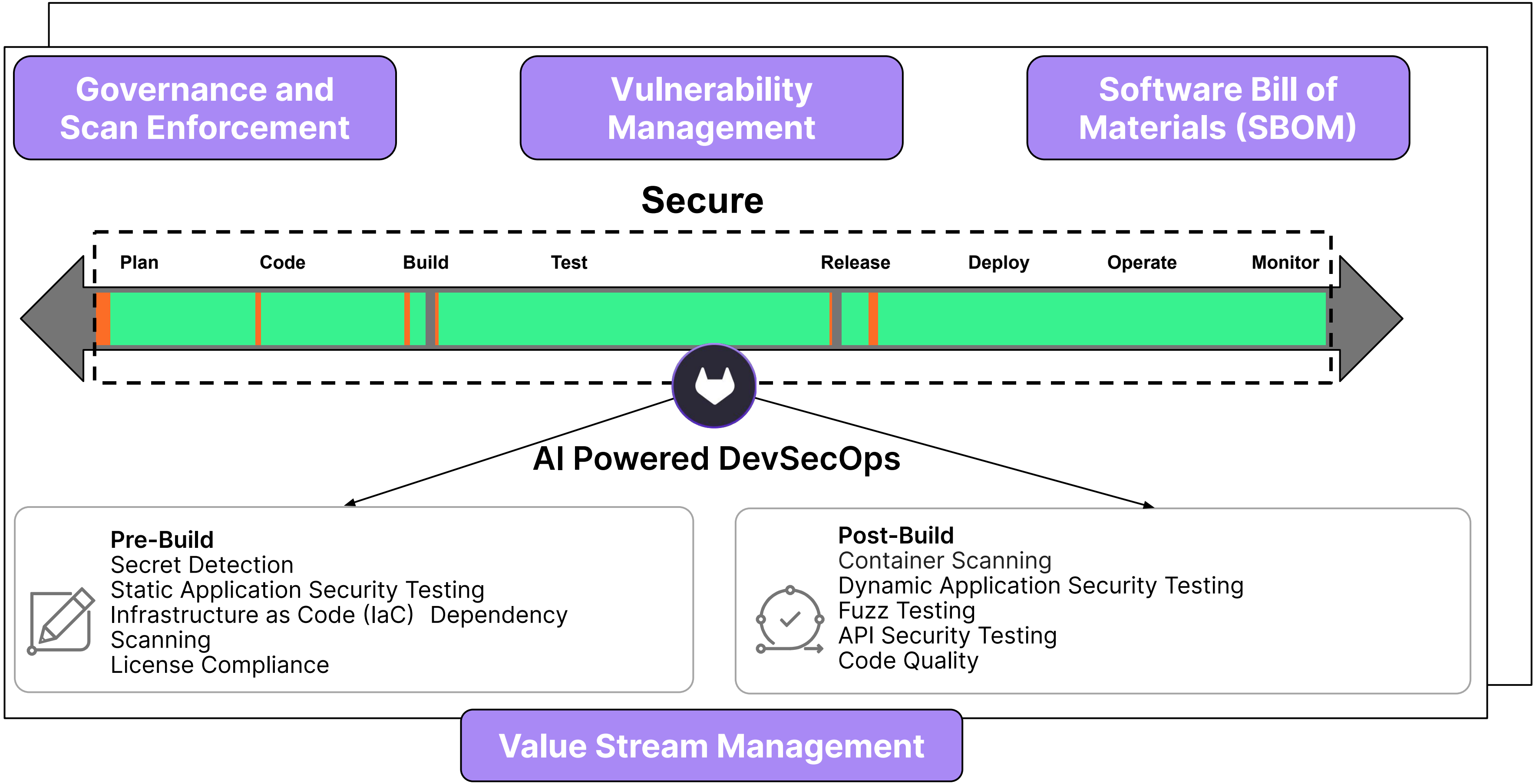


Pre-Build
Secret Detection
Static Application Security Testing
Infrastructure as Code (IaC) Dependency
Scanning
License Compliance

Post-Build
Container Scanning
Dynamic Application Security Testing
Fuzz Testing
API Security Testing
Code Quality

Value Stream Management

Securing The Speed - DevSecOps & VSM & AI



Governance and Scan Enforcement

Vulnerability Management

Software Bill of Materials (SBOM)

Secure

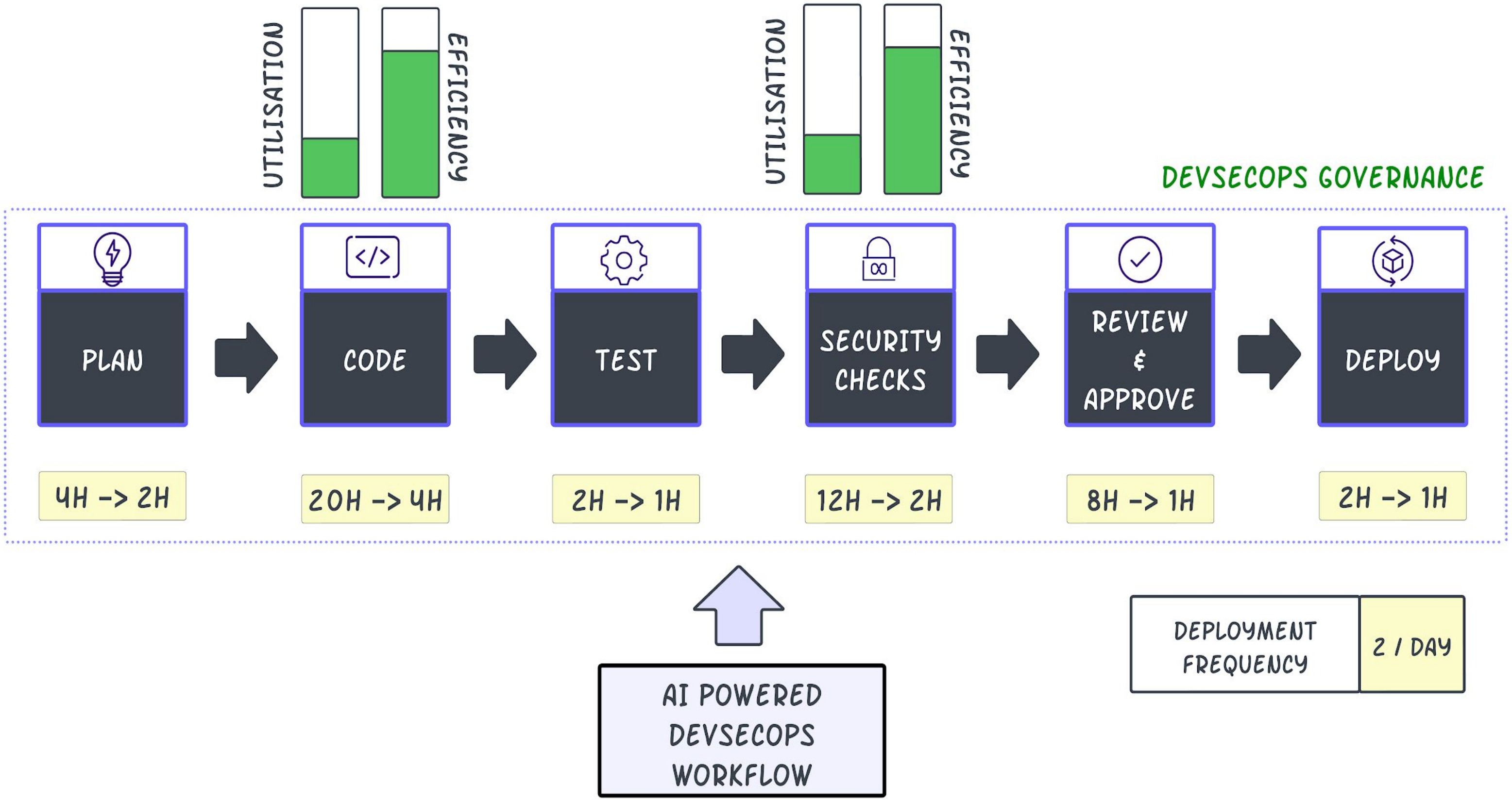
AI Powered DevSecOps

Pre-Build
Secret Detection
Static Application Security Testing
Infrastructure as Code (IaC) Dependency Scanning
License Compliance

Post-Build
Container Scanning
Dynamic Application Security Testing
Fuzz Testing
API Security Testing
Code Quality

Value Stream Management

Efficient DevSecOps - Value Stream Example

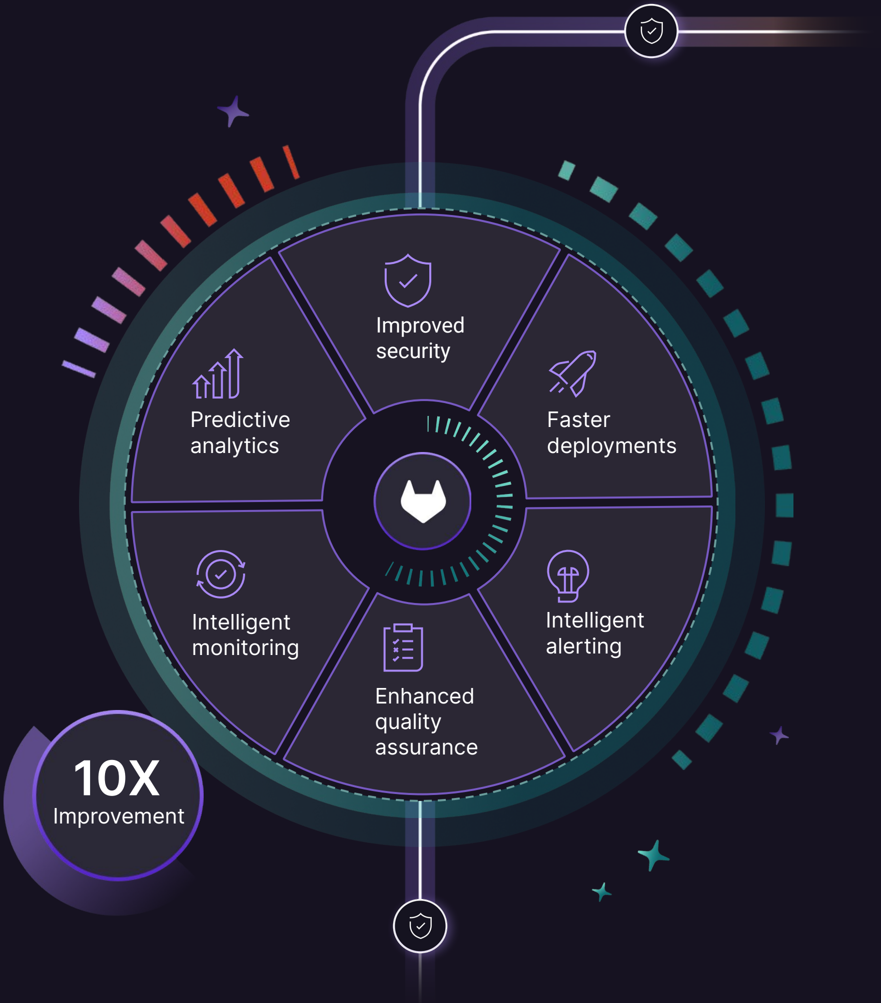


AI is central to GitLab's DevSecOps platform

Throughout the Software Delivery Lifecycle
Improve DevSecOps workflow efficiency by **10x** by applying AI assisted workflows to all teams involved in delivering software value

Privacy-First, Enterprise-Grade
Lead with a privacy-first approach allowing enterprises and regulated organizations to adopt AI assisted workflows

Single Application
Leverage the benefits of GitLab's single application to deliver more software faster, enabling executive visibility across value streams and preventing context switching

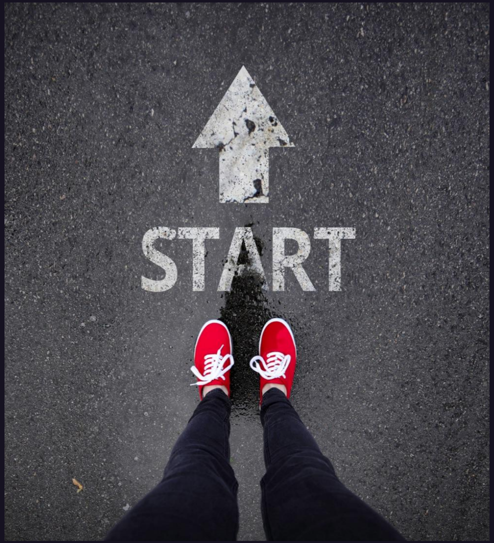


Call to Action

- What metrics do you currently use to identify DevSecOps (and business) bottlenecks and deficiencies?

- What does your DevSecOps Governance current state and future state look like?

- How do you track end-to-end visibility of the value streams in your organization?



GitLab

Thank you