GitLab

Integrating DevSecOps and Value Stream Management for AI-Driven Software Development Velocity

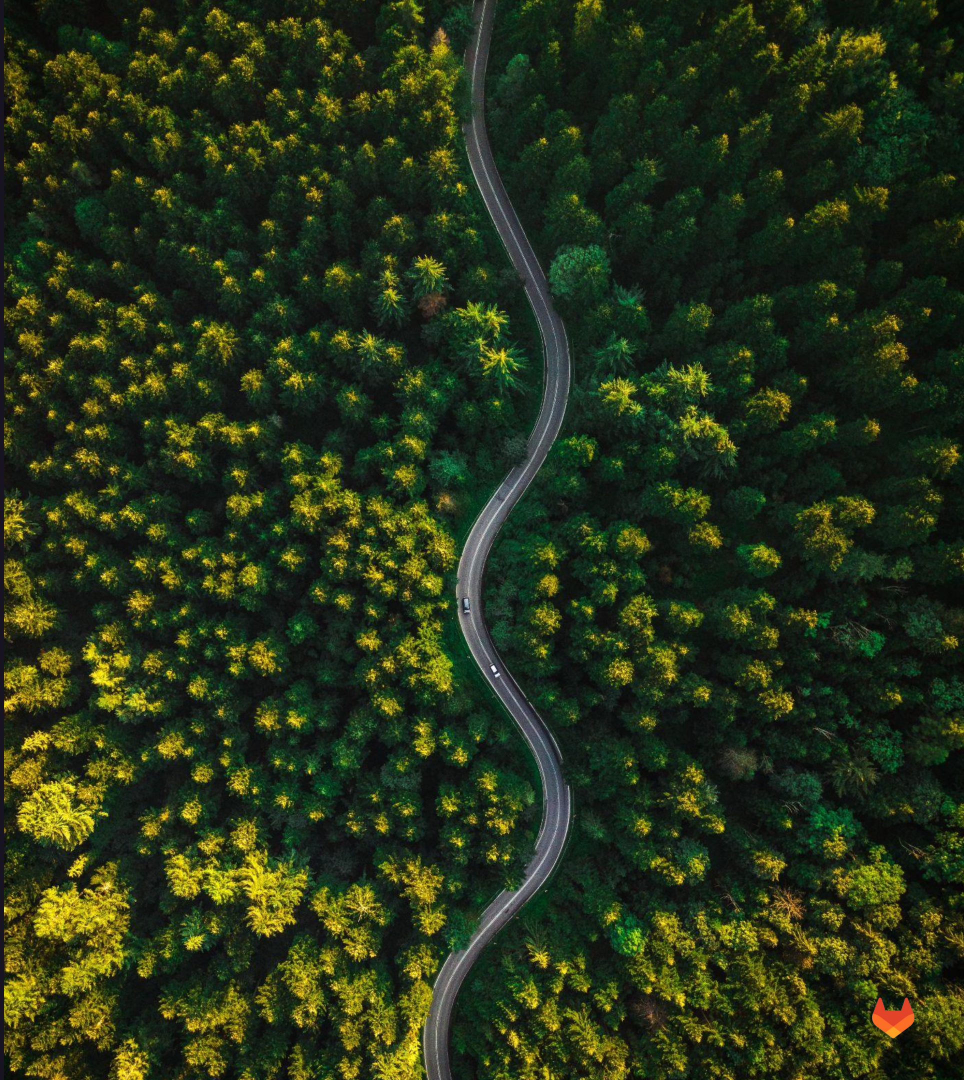
GitLab

Tomasz Skora

Staff Solutions Architect, APJ

GitLab

Introduction



The challenges we hear today

	Developer Awareness	Access Controls and Compliance	Toolchain TCO
Challenges	How do we improve the developer awareness on risk and remediation?	How do we prevent teams from bypassing security controls?	How do we decrease knowledge silos and improve collaboration?
Side effects	Increased Remediation Costs Compounding security debt	Complicated deployment approval Security coverage and chain of custody gap	Context switching Lack of end-to-end analytics Plateaued adoption
	Knowledge Silos	**Security Coverage Gap**	**Lots of Context Switching**

The cost of remediating security vulnerabilities

$59.5B
Annually cost of software bugs*

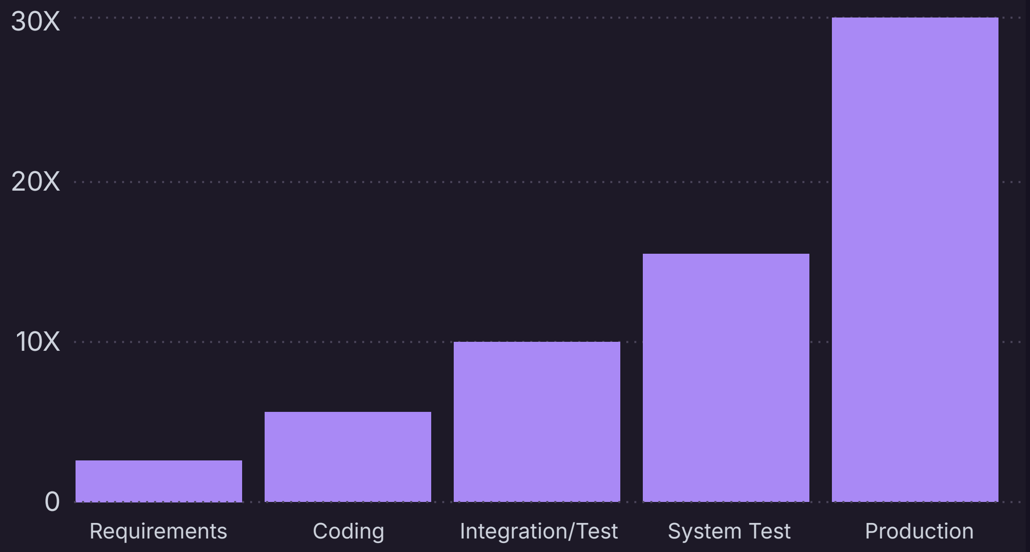
300
Cost of software developer hours**

Stage	Hours*	Cost
Coding stage	2.4	$740
Integration stage	4.1	$1,230
System stage	6.2	$1,860
Production stage	13.1	$3,930

*(NIST - Impact of Inadequate Software Testing
**2019 SW Dev Price Guide

Cost of Remediation

*X is a normalized unit of cost and can be expressed in terms of person-hours, dollars, etc.



Source: National Institute of Standards and Technology (NIST)

Too many tools and rise of AI undermine compliance and security at scale

57%

of security respondents said spending time maintaining many security tool makes it difficult to stay on top of **compliance**

40%

of security professionals were concerned that AI powered code generation will increase their **workload**

Source: GitLab 2023 DevSecOps Report

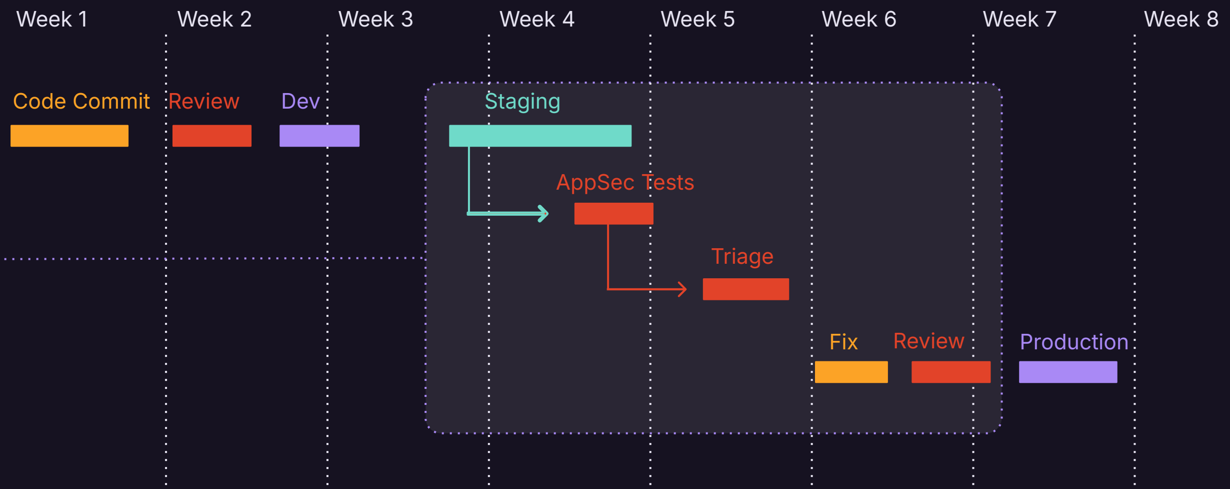
DevSecOps Value Stream

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*

Traditional AppSec Value Stream



Traditional AppSec Workflow

- Sequential
- Costly
- Loss of context

Knowledge Silos

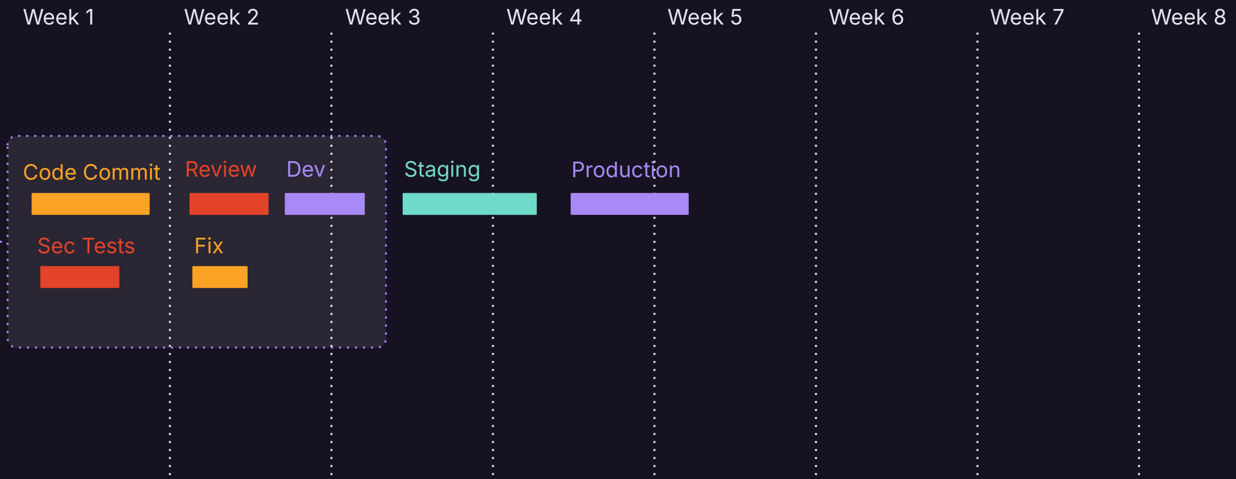
Security Coverage Gap

Lots of Context Switching

GitLab DevSecOps Value Stream

GitLab DevSecOps Workflow

- Embedded security
- Collaboration
- Speed
- Visibility



Improved coordination

Reduce feedback loops

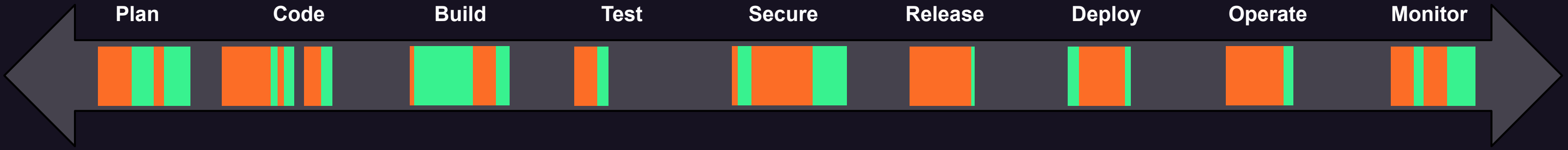
Increase product velocity

What is your current and future DevSecOps state?

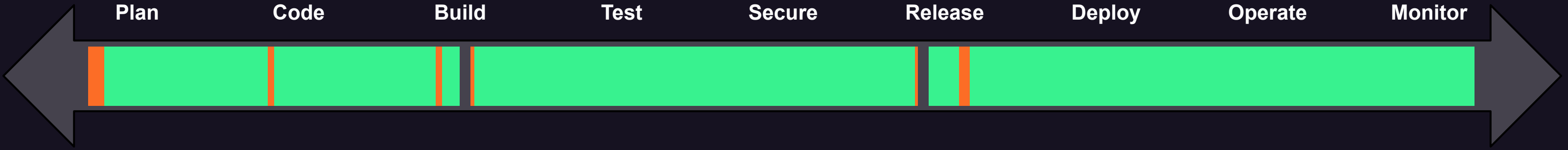
and how you measure your progress

- 🟩 Value Added Time
- 🟧 Non-Value Added Time
- ● Idle Time

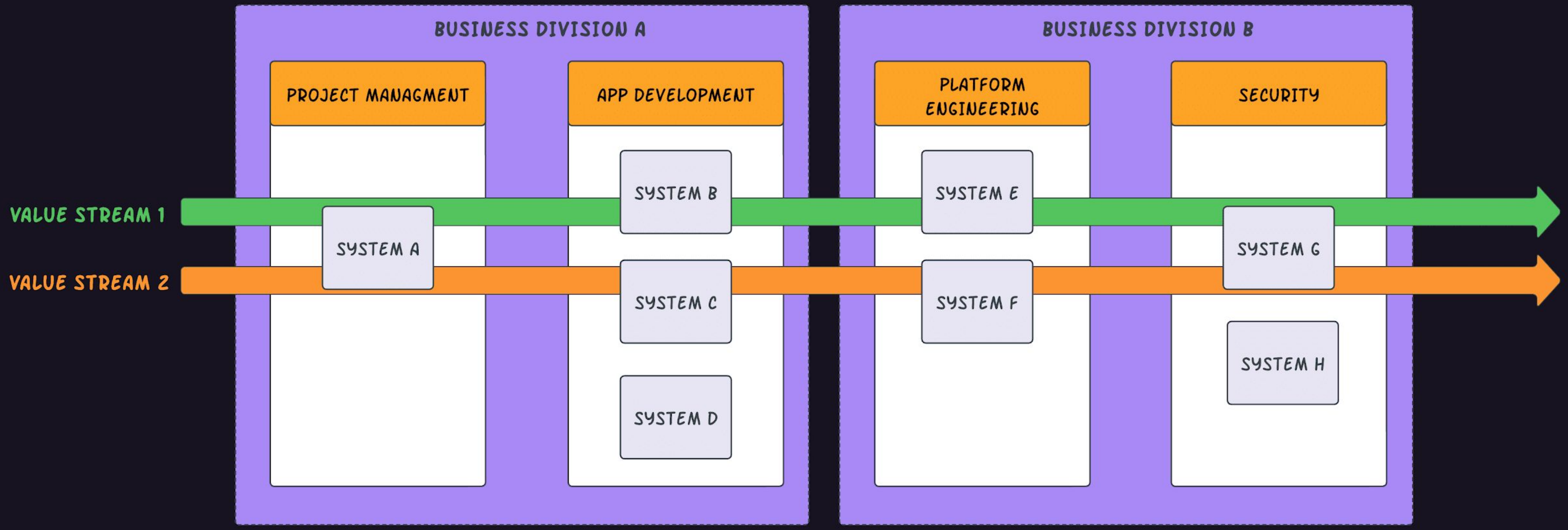
Current DevSecOps State



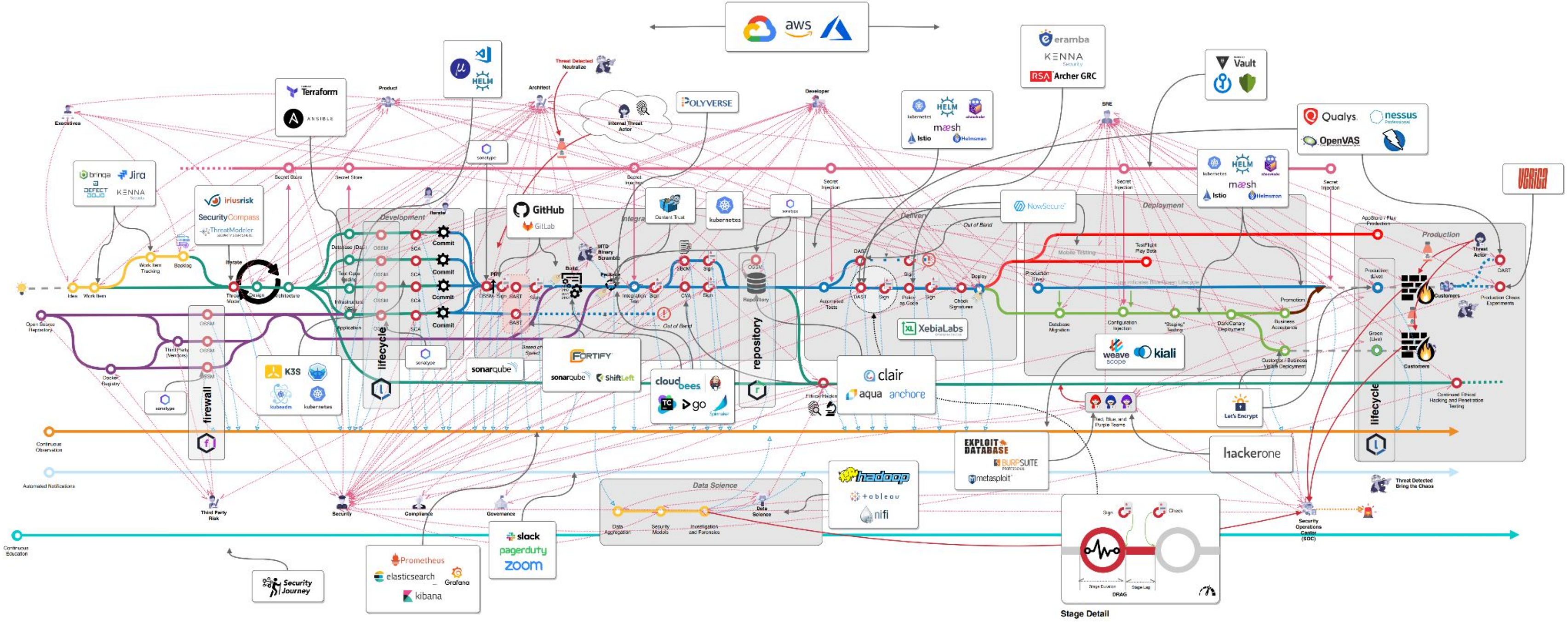
Desired DevSecOps Future State



Value Streams can be hard to measure ...



Visibility & Metrics & Governance ?



Source: Sonatype, Example DevSecOps Architecture

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

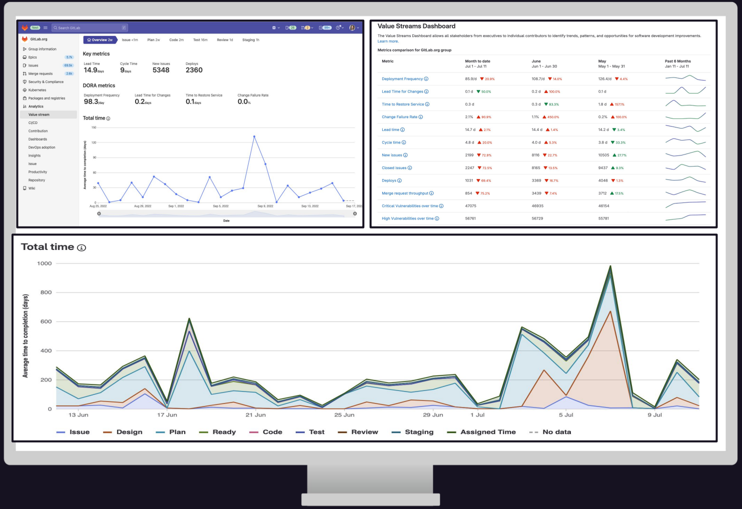
Measure

Visualise

Optimise

Value Stream Management enables executive visibility across value streams

- **Value streams dashboards** and metrics to identify security bottlenecks and deficiencies
- **Holistic visibility** and platform approach allows allows security leaders to gain a comprehensive understanding of security performance
- **Improved collaboration** to align security goals with other teams



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 %

Value Streams Dashboard

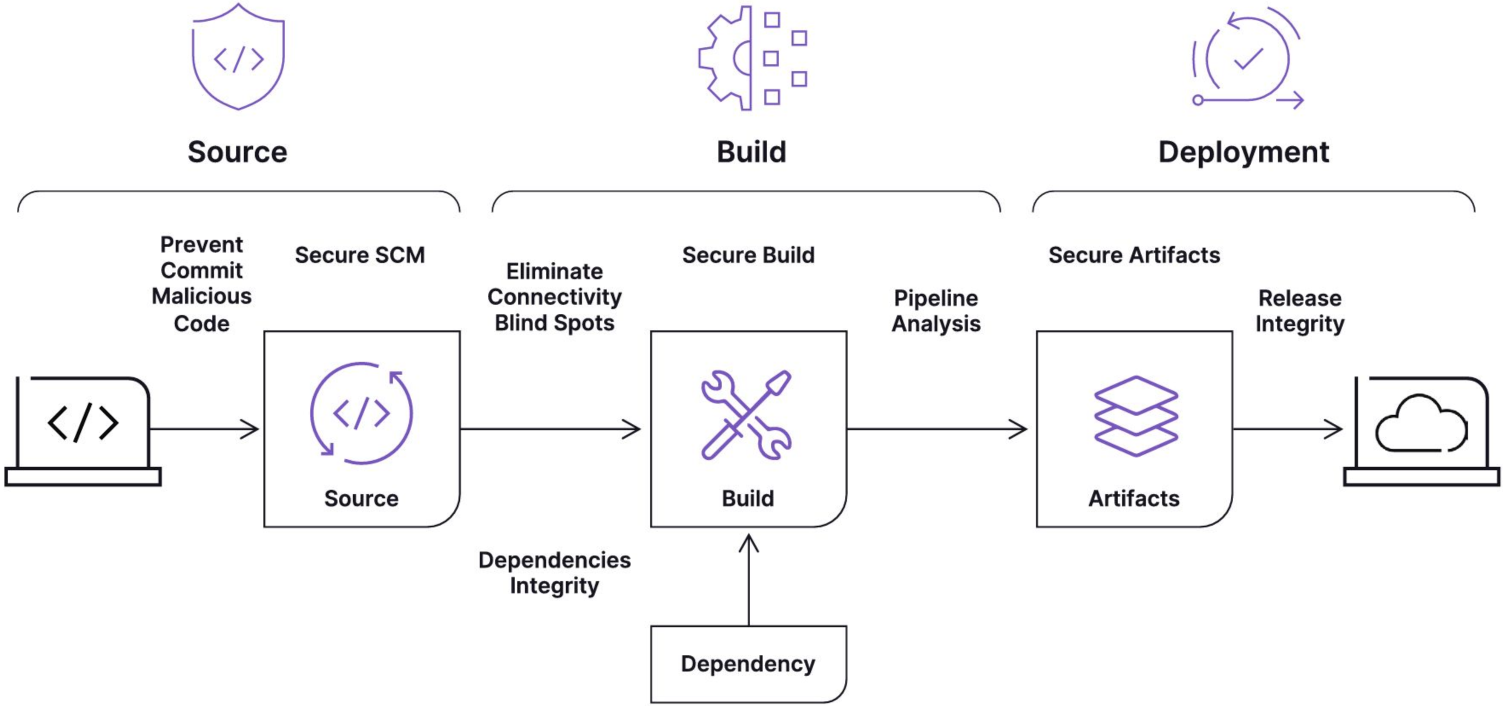
The Value Streams Dashboard allows all stakeholders from executives to individual contributors to identify trends, patterns, and opportunities for software development improvements. Learn more.

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	

Enhancing your Value Stream with DevSecOps Governance

Software Supply Chain Security Threats



Security was the **#1 Investment Priority** for companies in GitLab 2024 DevSecOps Annual Survey

Identify Vulnerable Software

All security tests embedded in the CI/CD pipeline using language-agnostic templates for standardised implementation

Contextual results within the MR to streamline remediation and review

BYOT to create a single pane of glass within the platform

SAST
Scan application source code and binaries

DAST
Analyze web applications for runtime threats

Dependency Scanning
Analyze external dependencies

IaC Scanning
Scan infrastructure misconfigurations

Secret Detection
Check for credentials in code commits

Container Scanning
Identify OS packages and dependencies

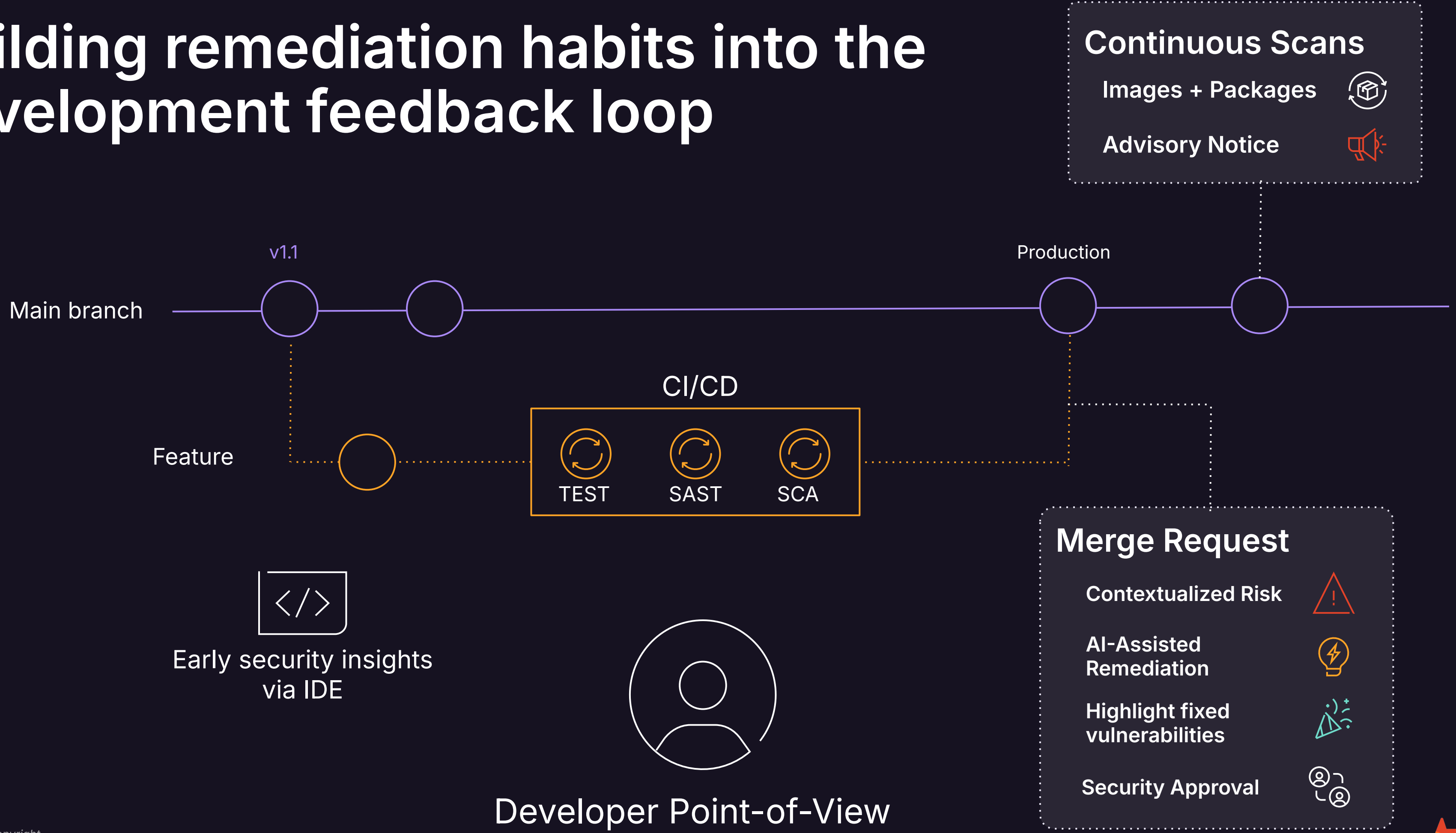
API Security
Analyze APIs for runtime vulnerabilities

Fuzz Testing
Use malformed data to measure app stability

Bring your own Tool
Easily integrate your existing security tools

Extract More Value
Use GitLabs policies, reporting, and guardrails

Building remediation habits into the development feedback loop



Continuous Scans

Images + Packages

Advisory Notice

v1.1

Production

Main branch

CI/CD

Feature

TEST

SAST

SCA

Merge Request

Contextualized Risk

AI-Assisted Remediation

Highlight fixed vulnerabilities

Security Approval

Early security insights via IDE

Developer Point-of-View

GitLab Copyright

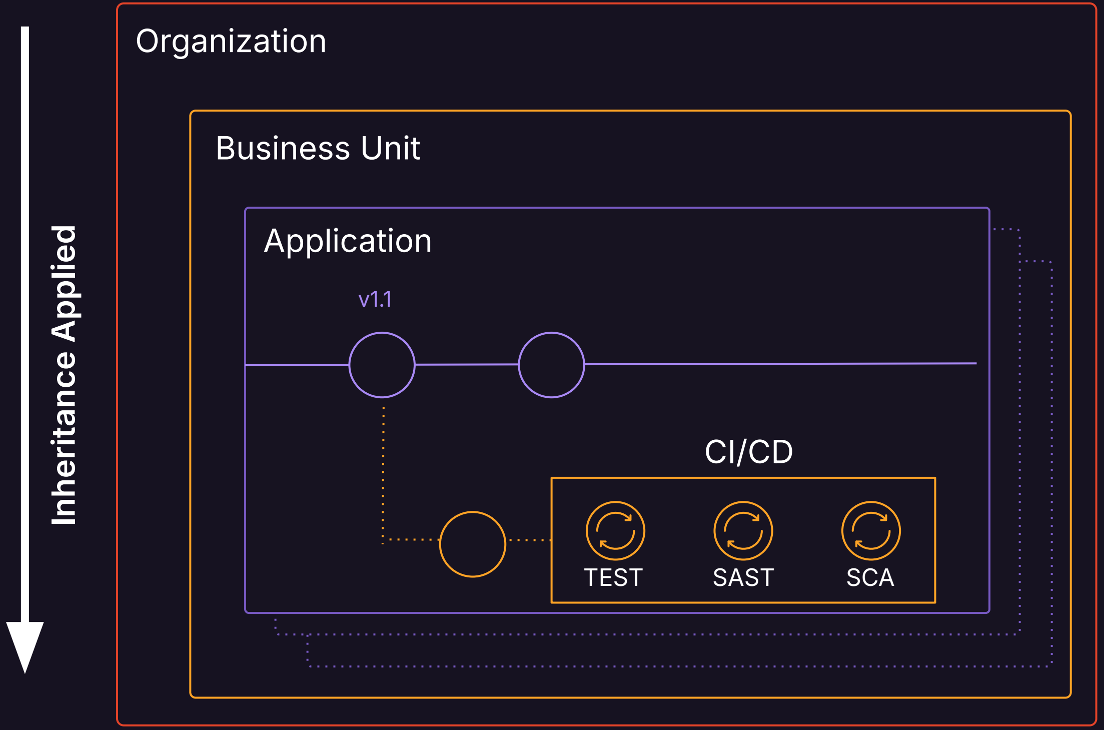
Scaling DevSecOps and Governance across the entire organization

Policy Enforcement

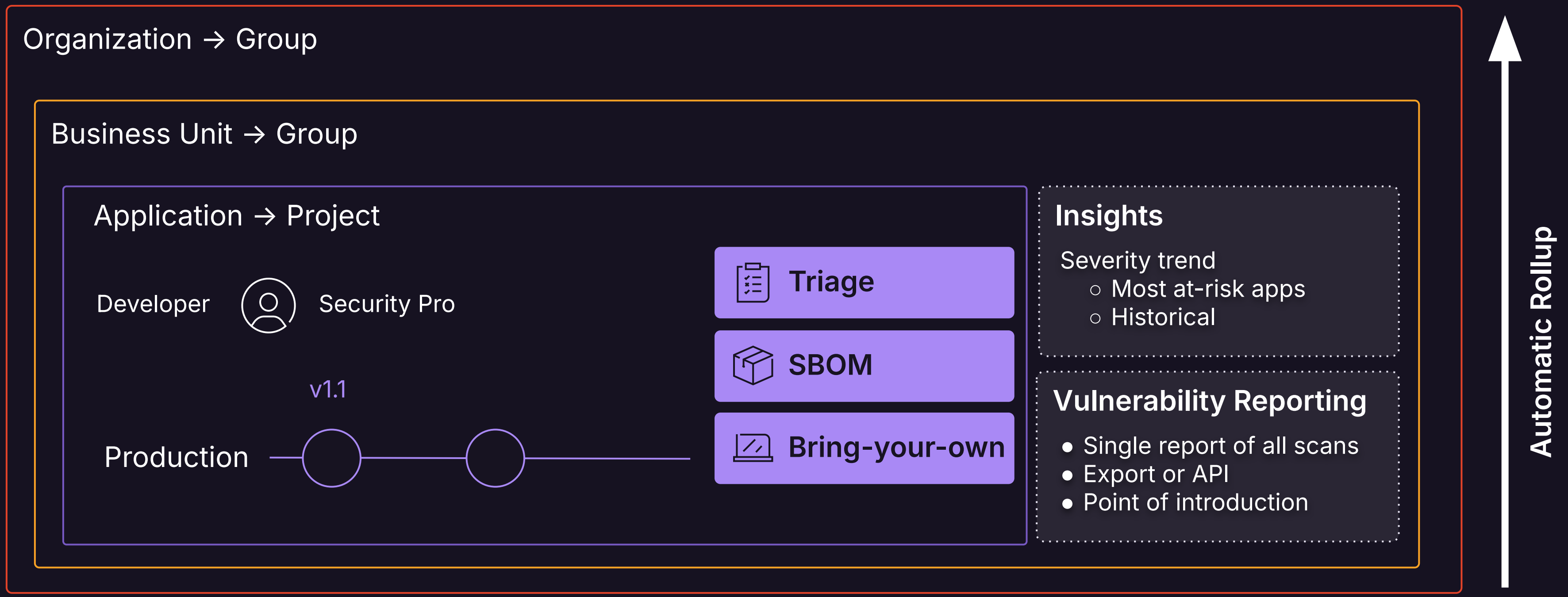
What severity threshold should be allowed into production?

What applications or repositories are not scanned?

Complete Security Coverage



Assess and prioritize risk across the organization



Organization → Group

Business Unit → Group

Application → Project

Developer

Security Pro

v1.1

Production

Triage

SBOM

Bring-your-own

Insights

Severity trend

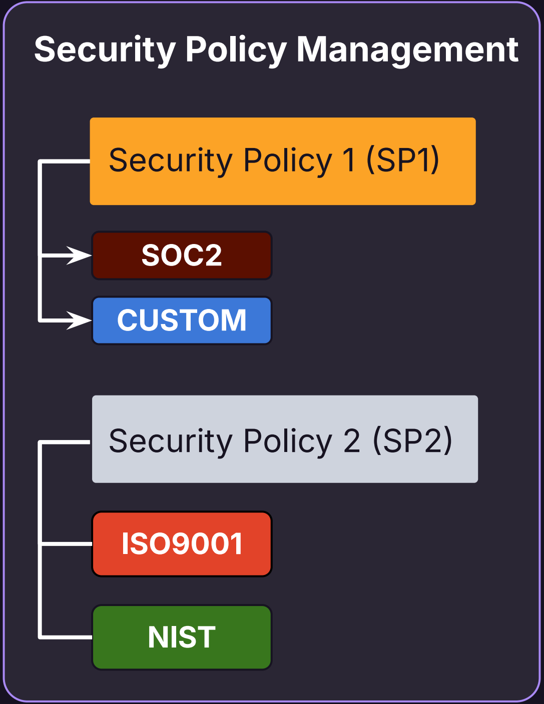
- Most at-risk apps
- Historical

Vulnerability Reporting

- Single report of all scans
- Export or API
- Point of introduction

Automatic Rollup

Simplifying DevSecOps Governance



Security Policy Management

Security Policy 1 (SP1)

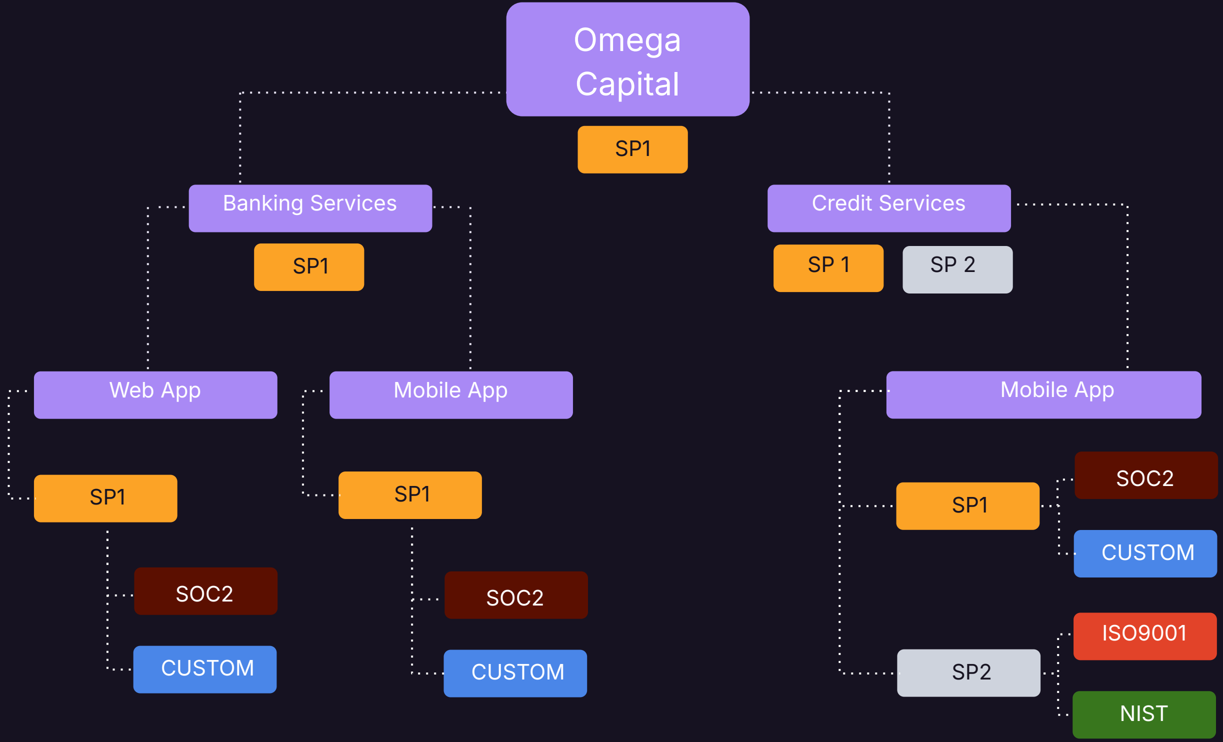
SOC2

CUSTOM

Security Policy 2 (SP2)

ISO9001

NIST



Omega Capital

SP1

Banking Services

SP1

Credit Services

SP 1

SP 2

Web App

SP1

SOC2

CUSTOM

Mobile App

SP1

SOC2

CUSTOM

Mobile App

SP1

SOC2

CUSTOM

SP2

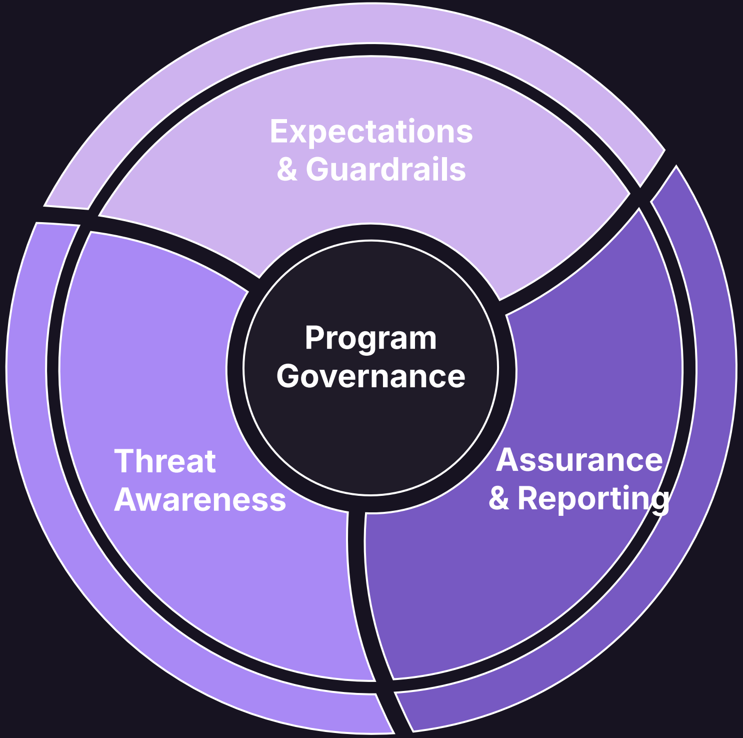
ISO9001

NIST

Establish a Scalable DevSecOps Program

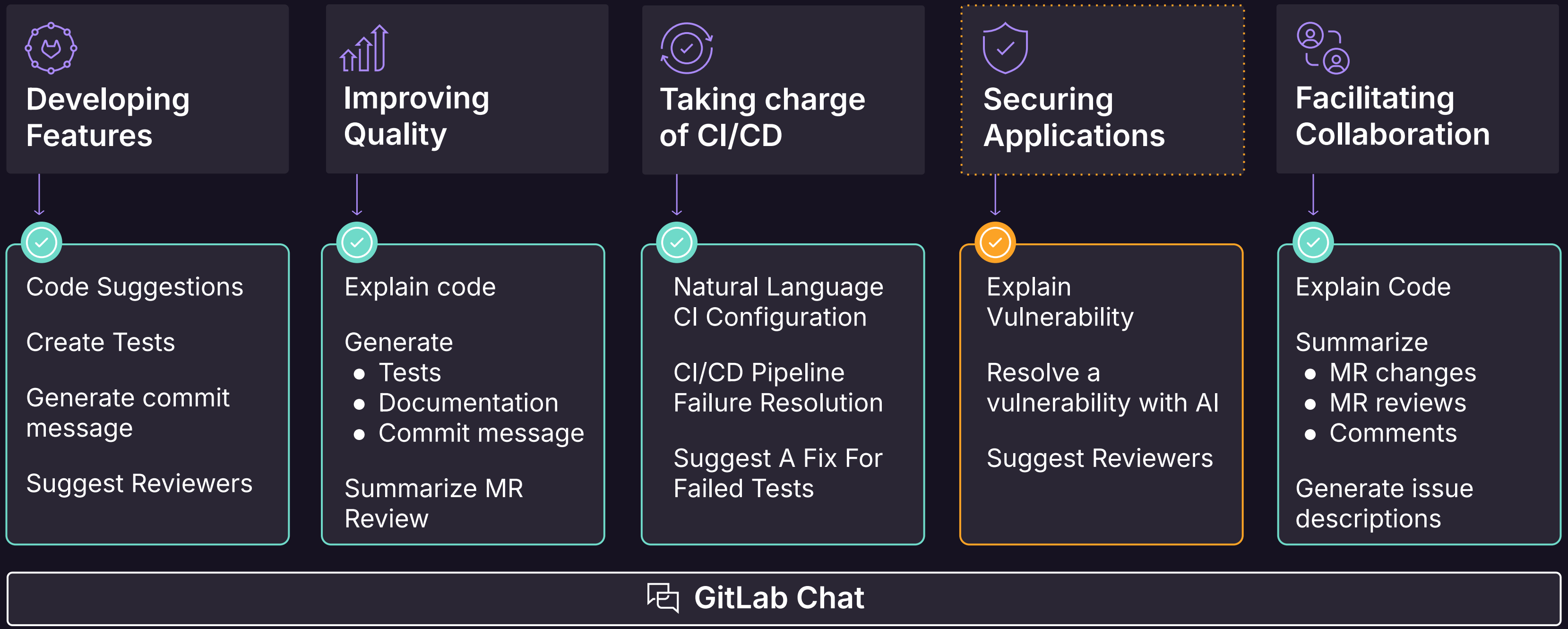
DevSecOps programs must:

- ✓ Give oversight and governance
- ✓ Allow creation of secure and efficient code
- ✓ Establish a secure software supply chain
- ✓ Enable consistent collaboration
- ✓ Improve time to market
- ✓ Be easily automatable



GitLab Duo+

AI-Powered DevSecOps workflows



Developing Features

Code Suggestions

Create Tests

Generate commit message

Suggest Reviewers

Improving Quality

Explain code

Generate
- Tests
- Documentation
- Commit message

Summarize MR Review

Taking charge of CI/CD

Natural Language CI Configuration

CI/CD Pipeline Failure Resolution

Suggest A Fix For Failed Tests

Securing Applications

Explain Vulnerability

Resolve a vulnerability with AI

Suggest Reviewers

Facilitating Collaboration

Explain Code

Summarize
- MR changes
- MR reviews
- Comments

Generate issue descriptions

GitLab Chat

How do you address these challenges in your DevSecOps Value Stream?

How do you improve developer awareness on contextual risk and remediation?

How do you prevent teams from bypassing security controls?

How do you decrease knowledge silos and improve collaboration?

GitLab

Thank you