

DevOps meet Sec

Your journey to deliver secure code fast

Davide Cioccia

13-02-2024

#whoami

Davide Cioccia

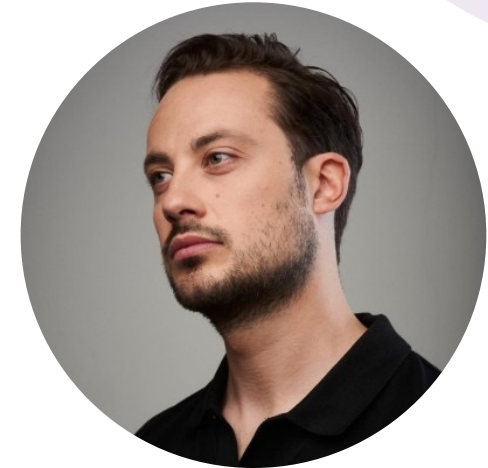
Senior Application Security Architect @ Veralto

DevSecCon Chapter Lead Netherlands

Speaker and trainer @ DEF CON, OWASP App Sec, BlackHat

Pentester / Security Engineer / Security Architect

Tennis and Padel player



davidecioccia



david3107



Google Trends

Home

Explore

Trending now



DevOps

Topic



+ Compare

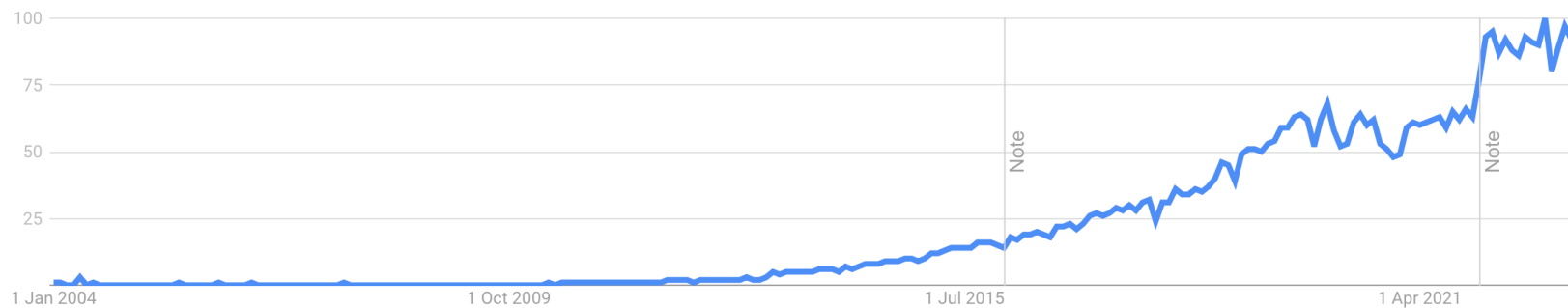
Worldwide ▼

2004 – present ▼

All categories ▼

Web Search ▼

Interest over time ?





● DevSecOps
Search term

+ Compare

Worldwide ▼

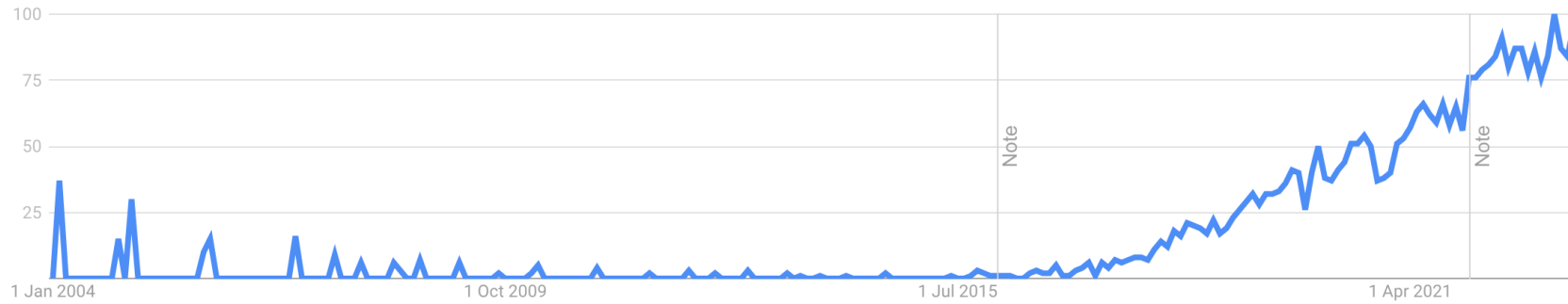
2004 – present ▼

All categories ▼

Web Search ▼

Interest over time ?

Help



DevOps
Topic

DevSecOps
Search term

+ Add comparison

Worldwide

2004 – present

All categories

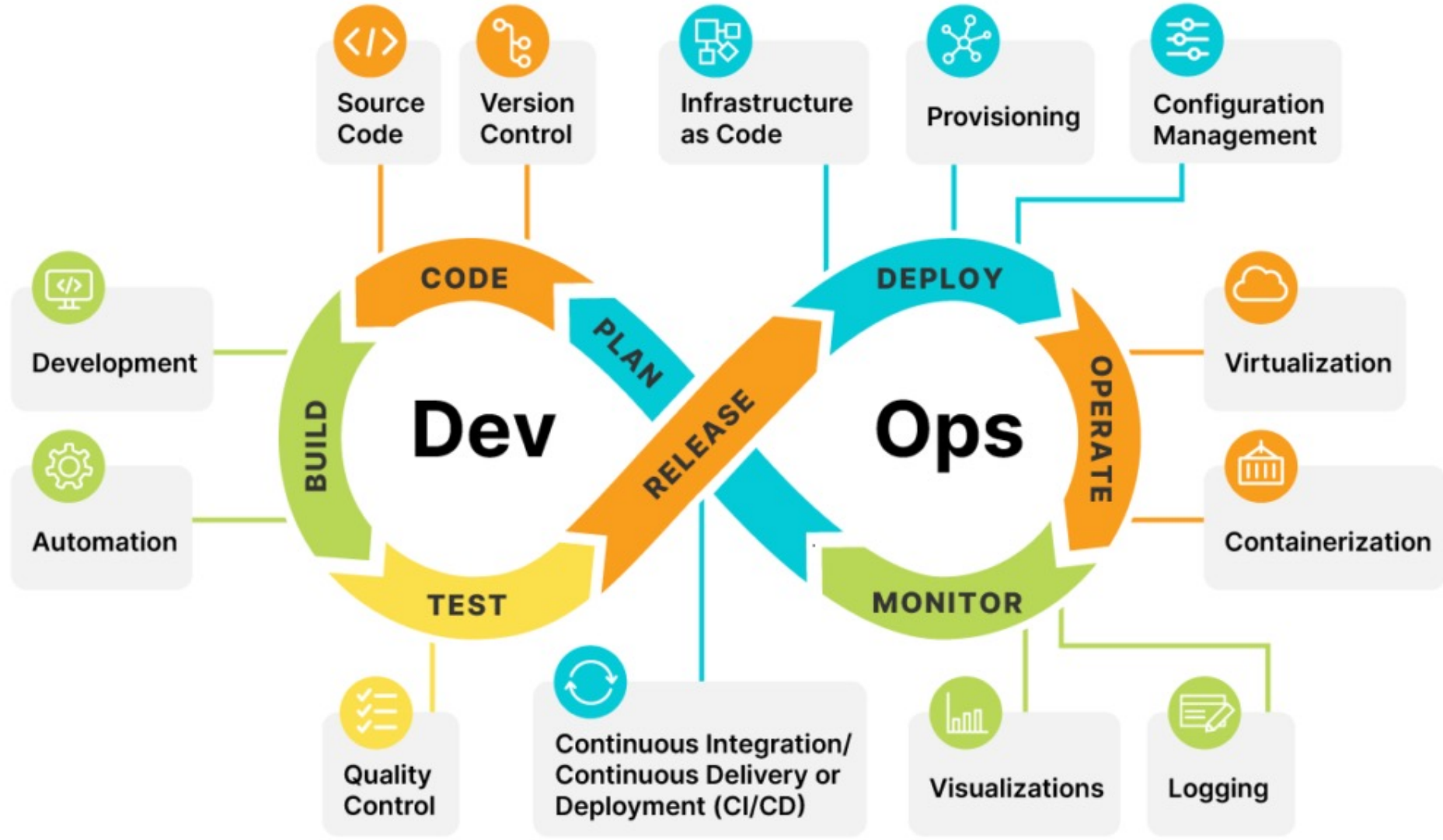
Web Search

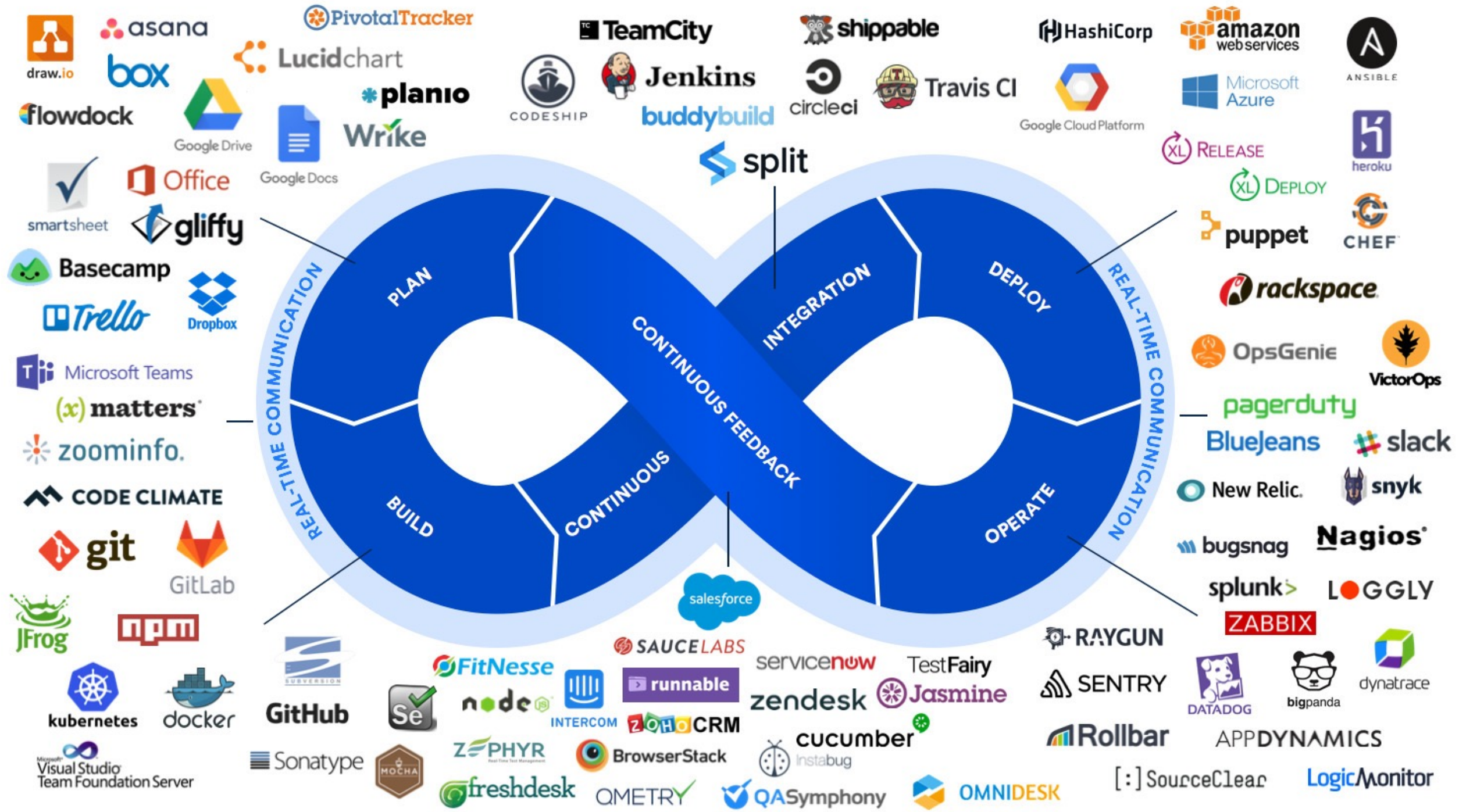
Note: This comparison contains both Search terms and Topics, which are measured differently.

[LEARN MORE](#)

Interest over time







Benefits of DevOps

- Faster, better product delivery.
- Faster issue resolution and reduced complexity.
- Greater scalability and availability.
- More stable operating environments.
- Better resource utilization.
- Greater automation.
- Greater visibility into system outcomes.
- Greater innovation.



Benefits of DevSecOps

- Faster, better product delivery.
- Faster issue resolution and reduced complexity.
- Greater scalability and availability.
- More stable operating environments.
- Better resource utilization.
- Greater automation.
- Greater visibility into system outcomes.
- Greater innovation.
- **Greater, faster and cheaper security**



What are the biggest challenges in software development in 2023?

We asked respondents to share, in their own words, their opinions on the biggest challenges in software development this year. Not surprisingly, security was a major theme. Here's what a few of the respondents had to say:

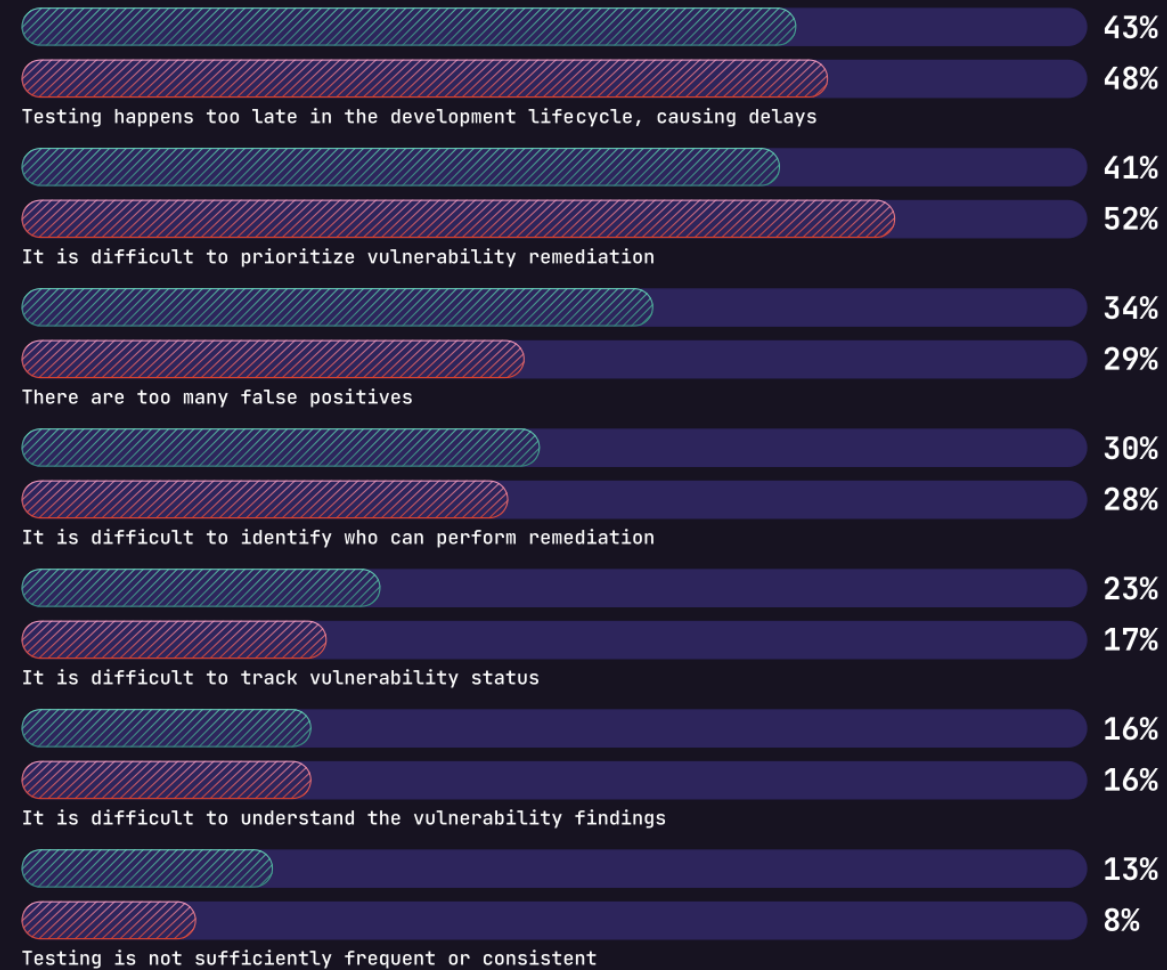
"Security, security, security, and more security... not only is this now an absolute MUST, we owe it to our customers, our organizations, our colleagues, ourselves, future DevOps Engineers, and humanity at large to do everything we can to create a safe, secure, compliant, and scalable future for our industry."

– DevOps Engineer, Healthcare

"There's too much focus from Product on pushing out new features without taking the time to keep an eye on **security, code quality, and code rot.**"

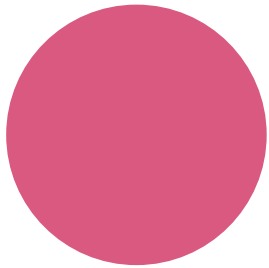
– Site Reliability Engineer, Media & Entertainment

Biggest frustrations with security testing, according to Security



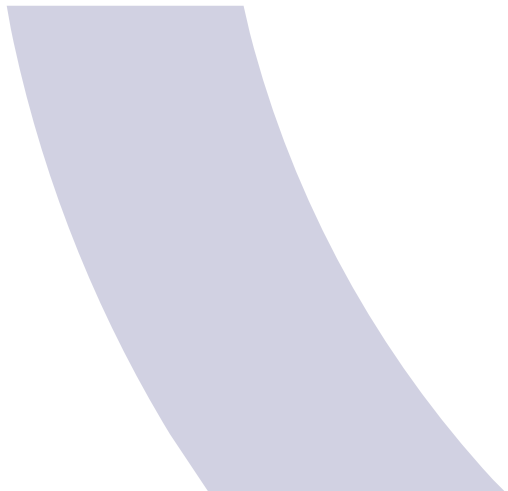
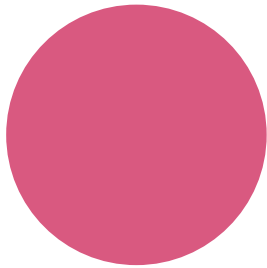
2023 2022





What I love about DevOps is that in the end, it's all about people.

Polar Squad · November 4, 2020



What I love about DevOps is that in the end, it's all about people.



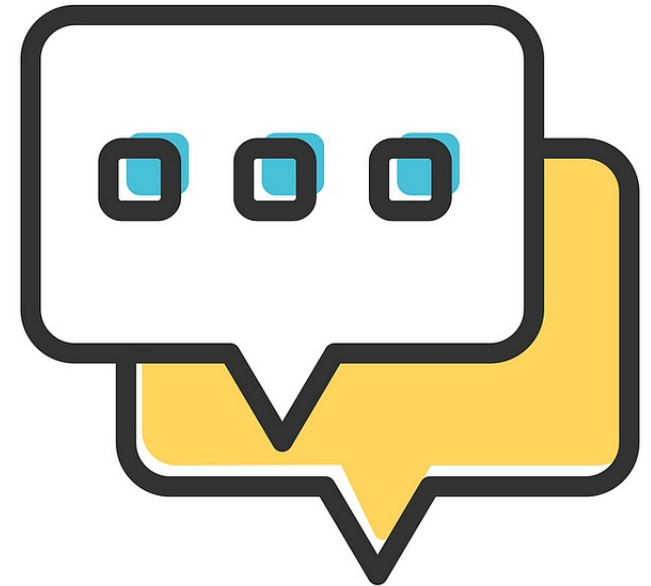
Polar Squad · November 4, 2020

CULTURE

What is CULTURE?

- How does the security culture look like in your company?

let's discuss



Poor security culture

- 1 security person for 1000 developers
- The security department owns the security of the product
- Security approves every change before production
- Rely only on a compliance pentest done once a year
- Management is not engaged in security discussions

Strong security culture

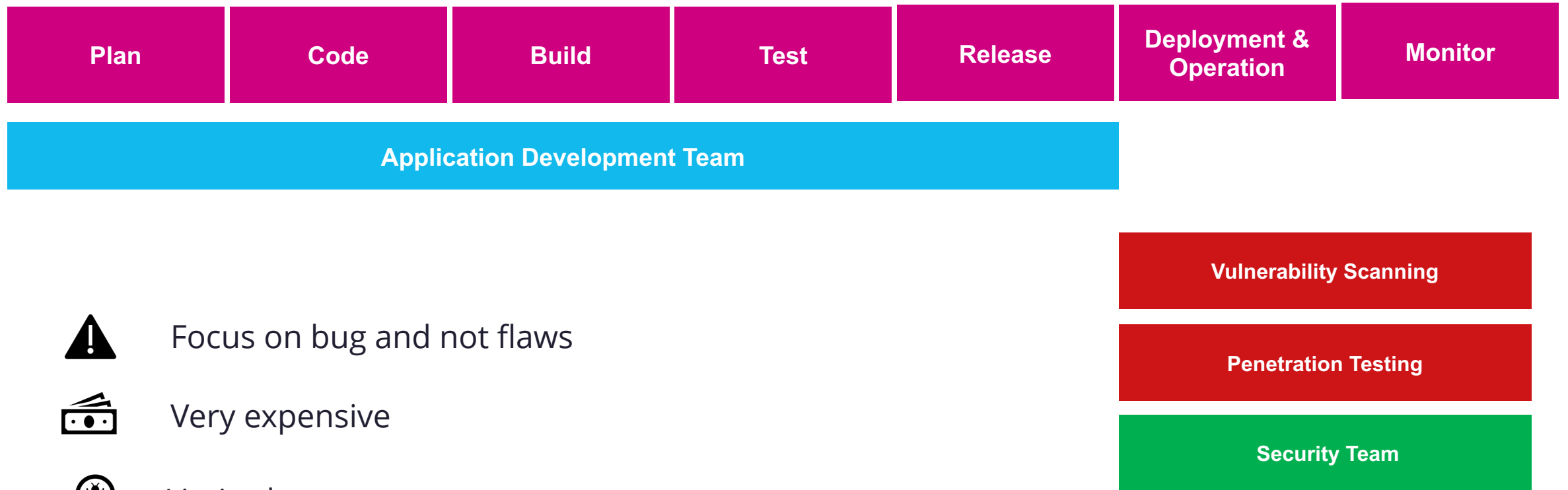
- Bring in security with development background
- Let security work together with the development team
- Security must understand the product tech and the business
- Development teams own the security of their products
- Management speaks regularly about security



#1 Pitfall: Security is not part of development

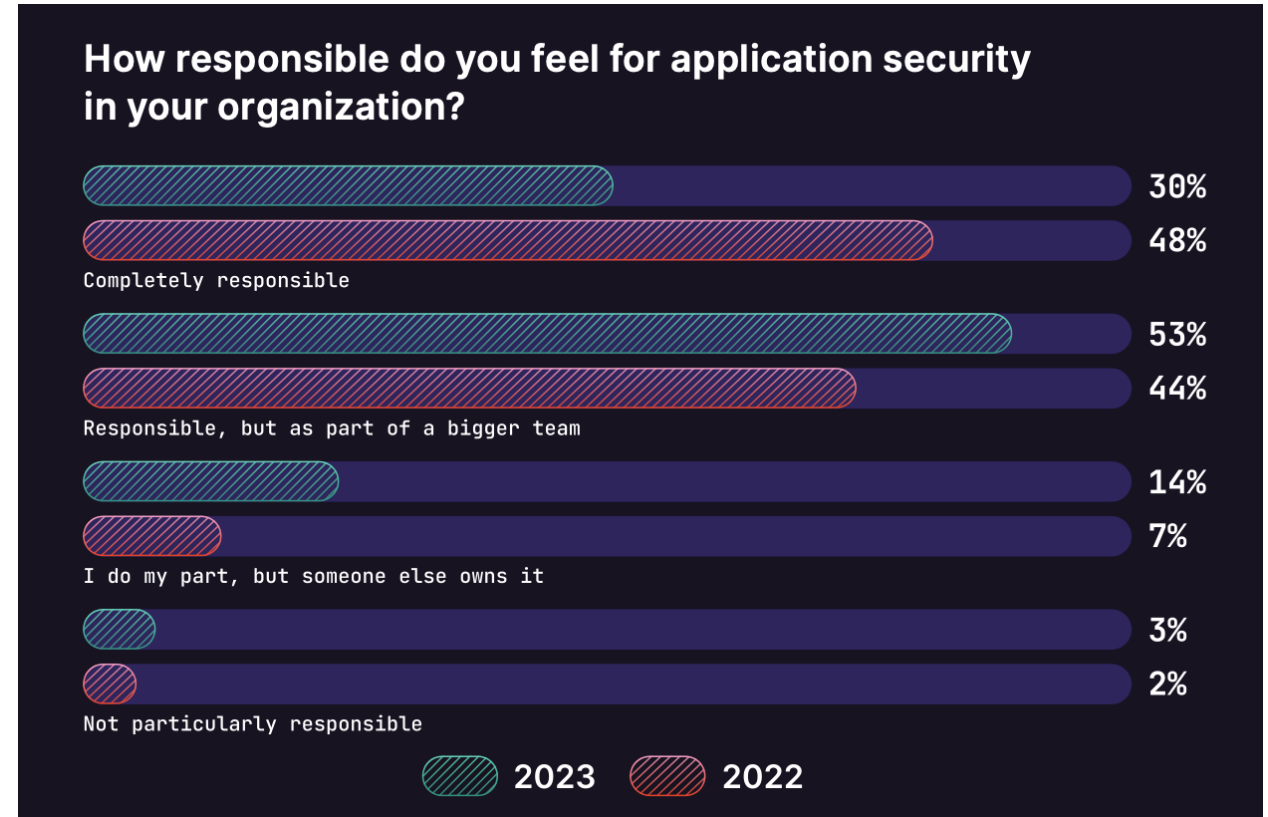


Pitfall #1: Security as afterthought



Better: Development and security work together

- Implement security in different phases of the development cycle
- Developers know their product best
- Security must know coding
- Teach security to developers and development to security

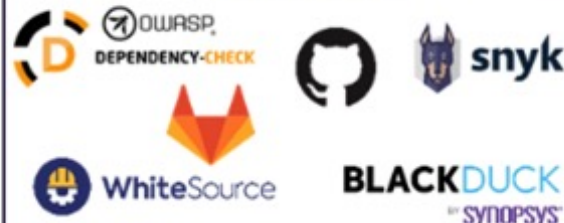


Pitfall #2: All the tools in one shot

Secret detection



SCA tools



SAST tools



Vulnerability Management



Security in IaC



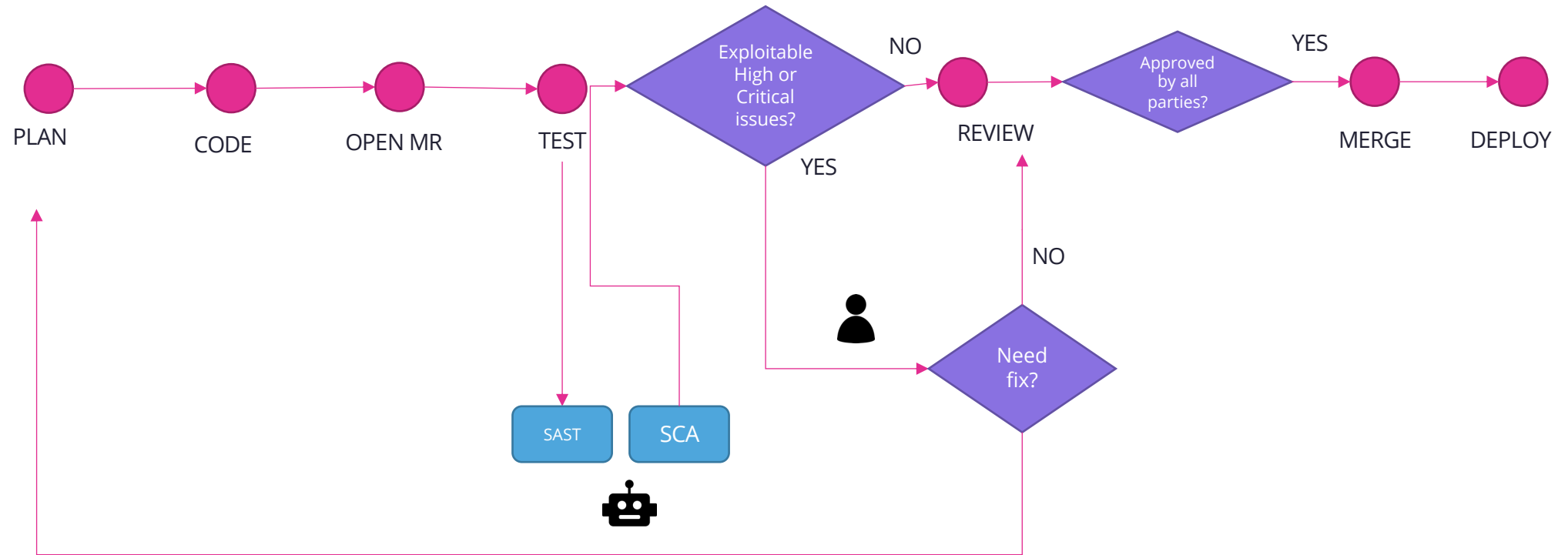
DAST tools



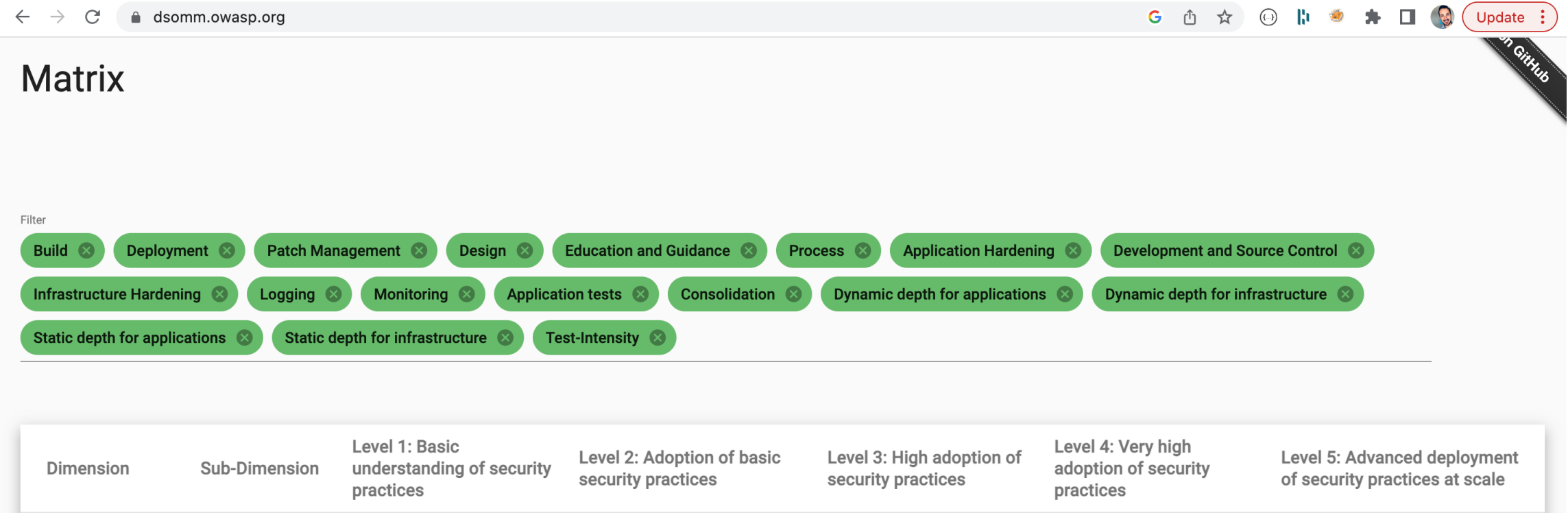
Secrets Management



Solution: Start small: SAST + SCA




Follow a Maturity Model (DSOMM)









































Pitfall #3: No issue tracker integration

Projects / Beyond Gravity

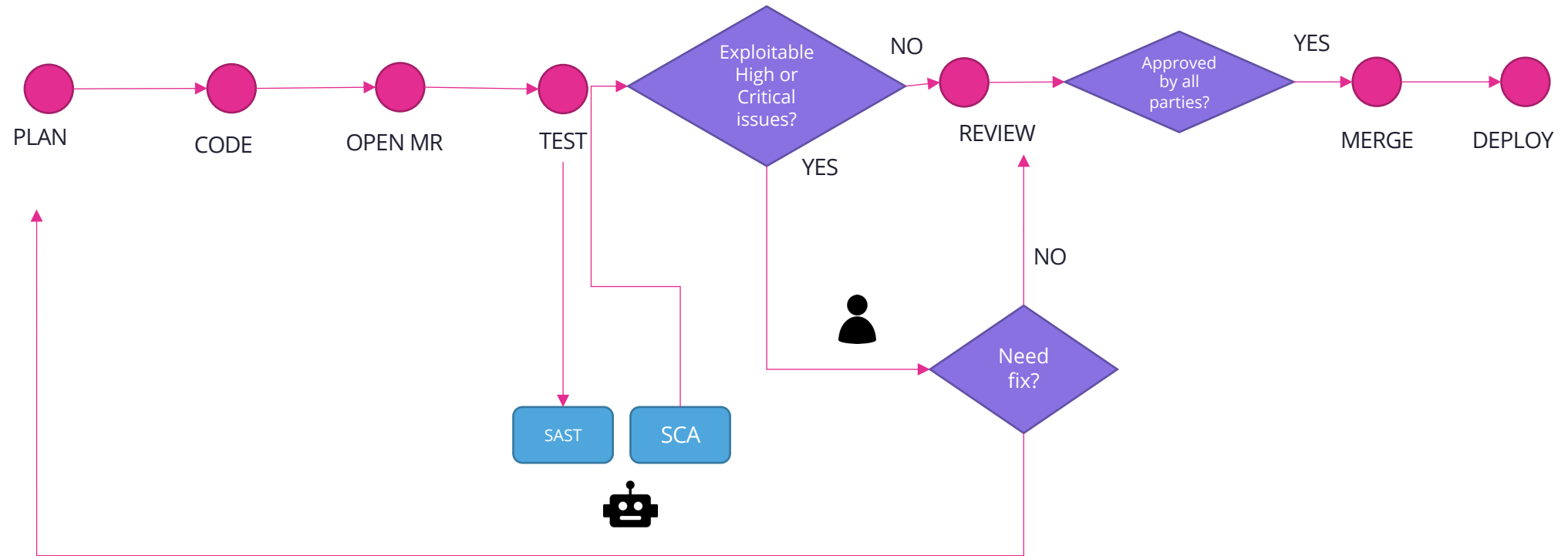
Board

Search:  +3 Epic ▾

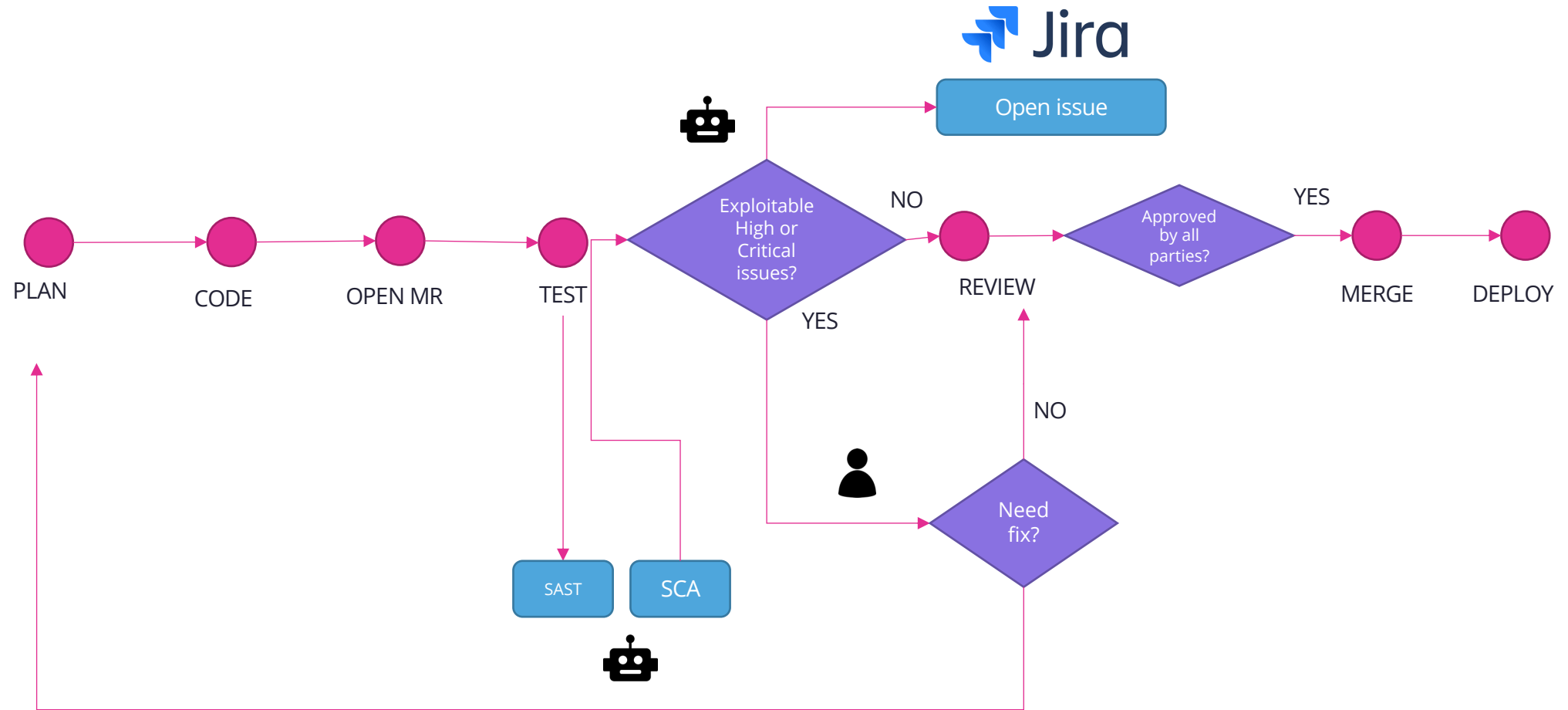
GROUP BY Choices ▾

TO DO 12	IN PROGRESS 4	IN QA 4	DONE 4
<p>Implement feedback collector</p> <p>NUC-205 9  </p>	<p>Update T&C copy with v1.9 from the writers guild in all products that have cross country compliance</p> <p>NUC-213  1  </p>	<p>Adapt web app no new payments provider</p> <p>NUC-346  5  </p>	<p>Quick booking for accommodations - web</p> <p>NUC-336   4  </p>
<p>Bump version for new API for billing</p> <p>NUC-206 3  </p>	<p>Bump feedback icon version</p> <p>NUC-214 3  </p>	<p>Purchasing error - edit fields</p> <p>NUC-354  3  </p>	<p>Fluid booking on tablets</p> <p>NUC-343  5  </p>
<p>Add NPS feedback to wallboard</p> <p>NUC-208 1  </p>	<p>Tech spike on new stripe integration with paypal</p> <p>NUC-215 3  </p>	<p>Multi-dest search UI web</p> <p>NUC-338 5  </p>	<p>Shoping cart purchasing error - quick fix required.</p> <p>NUC-354  1  </p>
<p>Add analytics events to pricing page</p> <p>NUC-209 3  </p>	<p>Change phone number field type to 'phone'</p> <p>NUC-217  1  </p>		
<p>Resize the images for the upcoming campaign</p> <p>NUC-210 1  </p>			

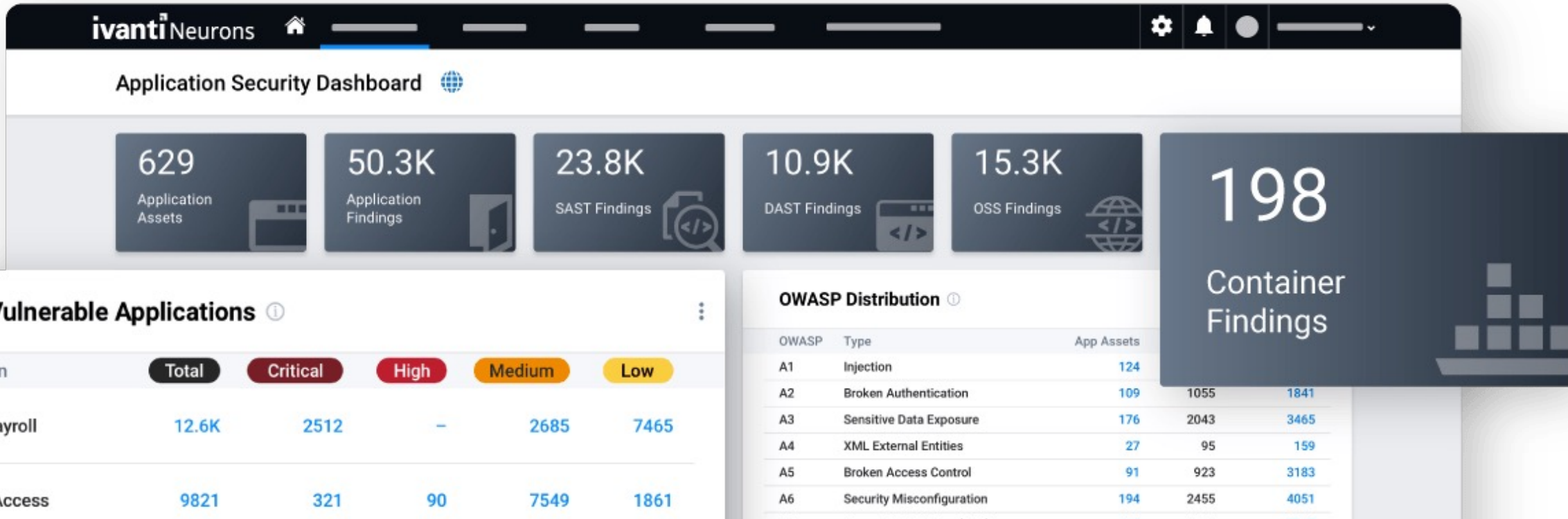
Integrate issues in development tools



Integrate issues in development tools



Pitfall #4: False positives



Focus on high risk and high probability issues

- Only scan production code. Remove test and debug code
- Focus on high fidelity findings
- Start with manual scans if you have never done a scan before and reduce noise
- Integrate automated scans as soon as there is no or very little noise

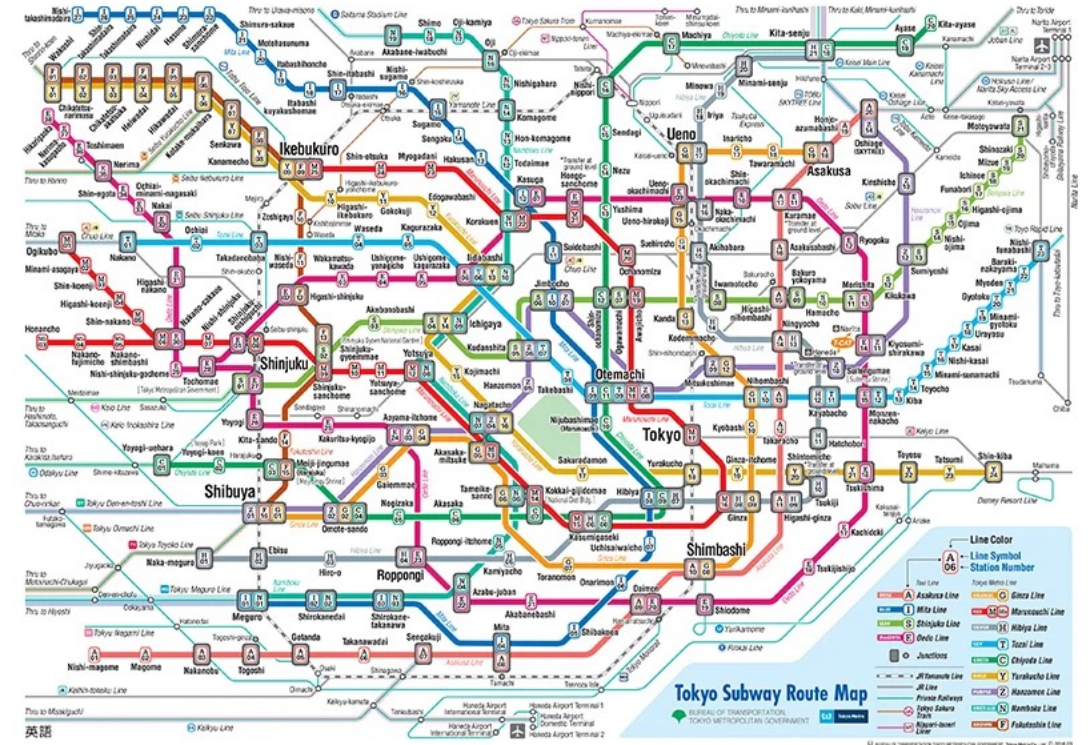
Pitfall #5: ONE security pipeline

- Do not define one security pipeline for all teams
- Not all the teams have the same maturity
- Products have different release cycles
- Product complexity will impact your speed
- Repo structure will need tool customization
- Branching strategy is different per team



Find the best fit for your pipelines

- Adopt a risk-based approach
- Secret scans for every change is great and fast
- SAST on every change is not ideal
 - Consider MR on protected branch scanning
- Scanning only new code or scanning always the full project
 - choose your tools



Pitfall #6: Missing git security posture

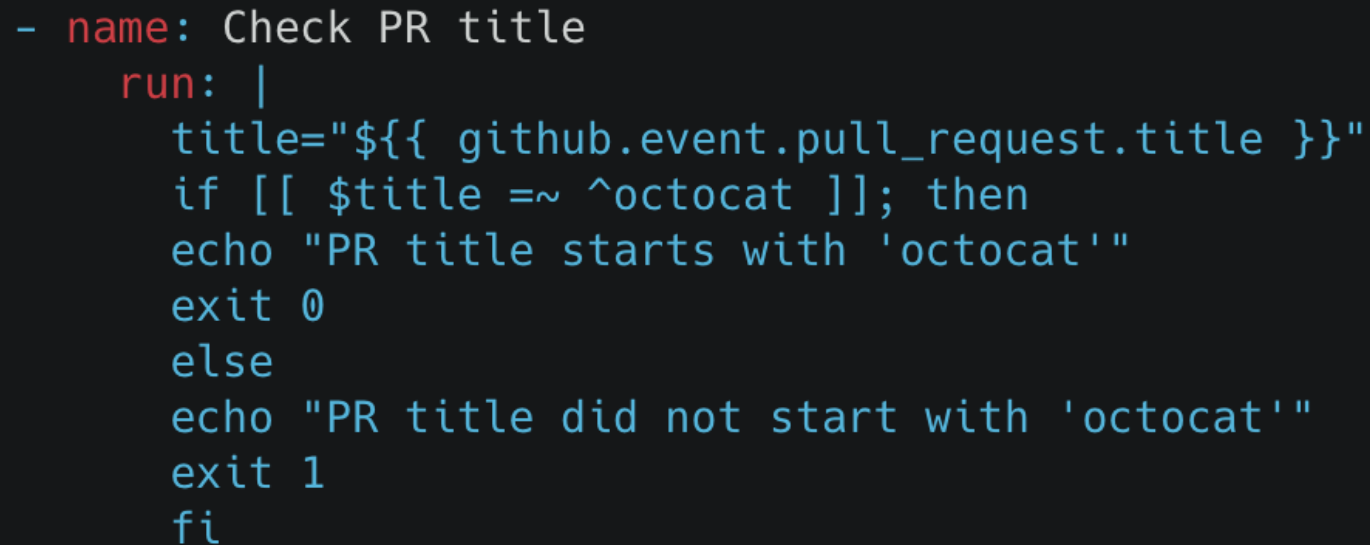
- You have selected the best code repository and versioning control system
- You have integrated the best security tools as part of your pipelines
- You detect issues from third party code, your code
- Do you control who can change your code?



Harden your version control and pipelines

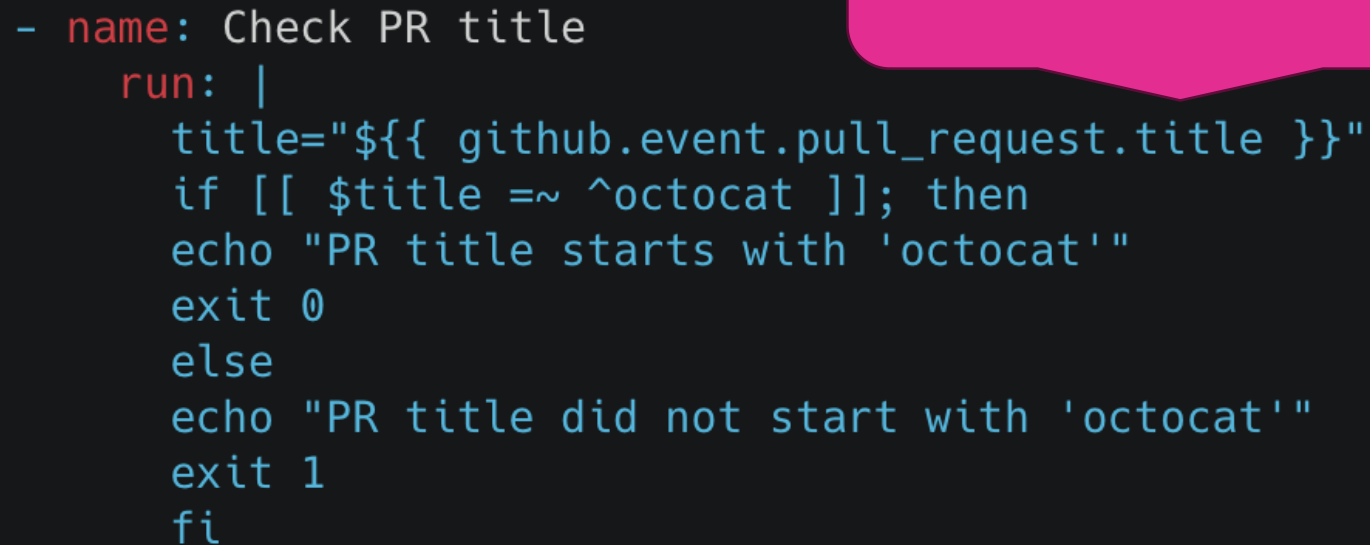
Check	Result	Severity
Repo has public visibility	repo is public	4
Branch protection is enabled for the default branch	enabled	✓ OK
Automatically dismiss approving reviews when someone pushes a new commit	not enabled	4
Blocks merging pull requests until code owners have reviewed	not enabled	4
Requires all conversations on code to be resolved before a pull request can be merged into a branch	not enabled	2
Require signed commits on protected branch	not enabled.	3
Enforce all configured restrictions for administrators.	not enabled	3
Prevents anyone from pushing merge commits to a branch.	not enabled	2
Permits force pushes to the protected branch	enabled	✓ OK
Allow deletion of the protected branch	enabled	✓ OK
Days since latest commit	last commit is older than 30 days.	4
Collaborators with admin privileges	not present	✓ OK
Collaborators with site_admin privileges	not present	✓ OK
High and Critical open issues	0	✓ OK

Harden your pipeline



```
- name: Check PR title
  run: |
    title="${{ github.event.pull_request.title }}"
    if [[ $title =~ ^octocat ]]; then
      echo "PR title starts with 'octocat'"
      exit 0
    else
      echo "PR title did not start with 'octocat'"
      exit 1
    fi
```

Harden your pipeline



```
- name: Check PR title
  run: |
    title="${{ github.event.pull_request.title }}"
    if [[ $title =~ ^octocat ]]; then
      echo "PR title starts with 'octocat'"
      exit 0
    else
      echo "PR title did not start with 'octocat'"
      exit 1
    fi
```

a"; ls \$GITHUB_WORKSPACE":

Harden your pipeline

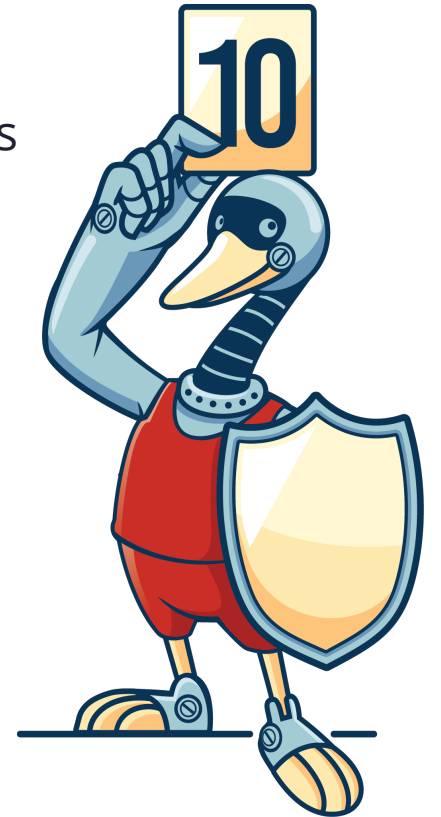
a"; ls \$GITHUB_WORKSPACE":



```
Run title="a"; ls $GITHUB_WORKSPACE""  
README.md  
code.yml  
example.js
```

Best practices

- Use actions or runners instead of run scripts
- Do not pass untrusted input (title, description, comments for example) to scripts
- Scan your pipelines for security vulnerabilities using SAST and SCA tools
- Restrict permissions for tokens
- Use OSSF Scorecard



Pitfall #7: Security as mandatory approver for MR



Review required

At least 1 approving review is required by reviewers with write access. [Learn more.](#)



All checks have passed

2 successful checks

[Show all checks](#)



Merging is blocked

Merging can be performed automatically with 1 approving review.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Adopt a risk based approach

- Security does not have to approve every change in the code
- Fail the pipeline based on your policy (Exploitable High and Critical from tools)
- Add security to the MR automatically when security knowledge is needed
- Adopt a model to identify “security sensitive changes”
 - OWASP ASVS checklists for example
 - AI bots can help automate this process

Risk Model

ASVS Chapter	Rating	ASVS Requirements			
		Passed	Failed	Issues	Hotspots ?
V1 - Architecture, Design and Threat Modeling	E	6	1	1	0
V2 - Authentication	D	31	8	2	22
V3 - Session Management	A	3	2	0	13
V4 - Access Control	A	3	1	0	22
V5 - Validation, Sanitization and Encoding	E	12	8	22	9
V6 - Stored Cryptography	A	4	9	0	24
V7 - Error Handling and Logging	A	8	0	0	0
V8 - Data Protection	A	9	0	0	0
V9 - Communication	A	4	2	0	11
V10 - Malicious Code	A	1	0	0	0
V11 - Business Logic	A	3	0	0	0
V12 - Files and Resources	E	8	3	6	3
V13 - API and Web Service	E	6	3	19	22
V14 - Configuration	A	6	0	0	0
ASVS Rating	E	122	19		

Pitfall #8: We do not do Threat Model



thaddeus e. grugq 🌻
@thegrugq



Your threat model is not my threat model.



What's the “problem”

- DevOps processes are too fast for threat model
- Threat Model is complex
- We always need security to do threat model
- We will do it once and we will forget about it
- Documentation is an old concept

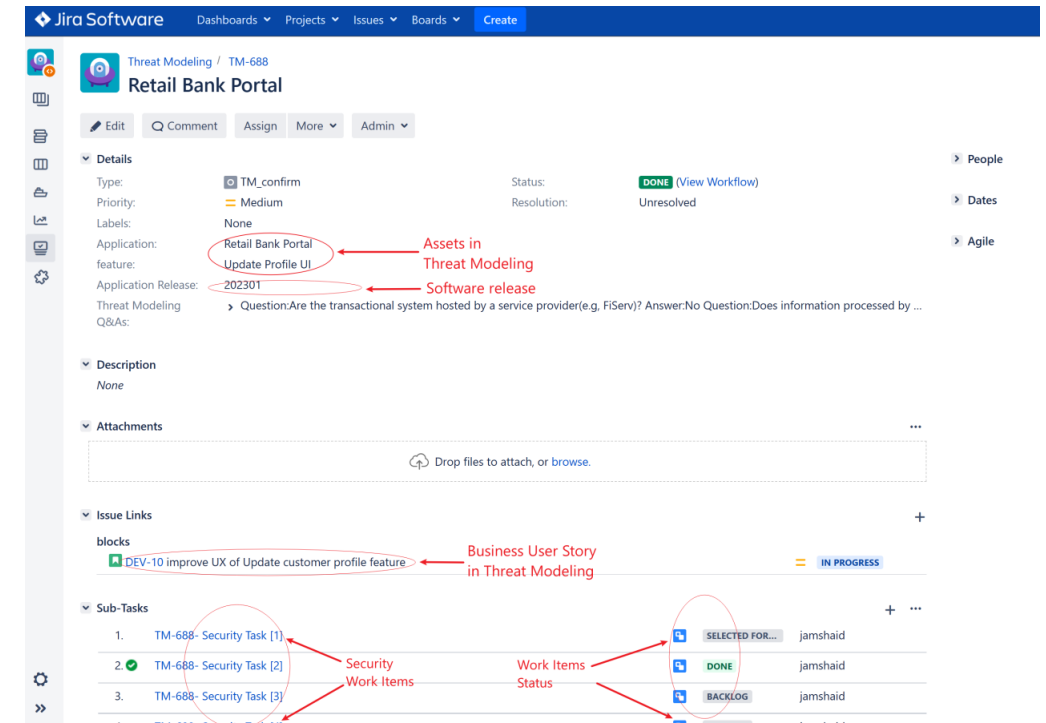
What's the “problem”

- DevOps processes are too fast for threat model
- Threat Model is complex
- We always need security to do threat model
- We will do it once and we will forget about it
- Documentation is an old concept

FALSE

How to do it in DevSecOps

- Adopt a model that will help you categorize and identify a set of threats
 - ask the security person to help ;)
- Do threat models in your planning tool (Jira, ServiceNow etc)
- AI can help speed up the process
- Discuss and log threats during your sprint planning
 - Feature X:
 - What can go wrong
 - What are we going to do to remediate it
 - How do we avoid forever



<https://marketplace.atlassian.com/apps/1229875/ai-assisted-threat-modeling-saas-powered-by-jira>

What are we looking for?

The 2021 OWASP Top 10 list

A01:2021

Broken
Access Control

A02:2021

Cryptographic
Failures

A03:2021

Injection

A04:2021

Insecure Design

A05:2021

Security
Misconfiguration

A06:2021

Vulnerable
and Outdated
Components

A07:2021

Identification
and Authentication
Failures

A08:2021

Software and
Data Integrity
Failures

A09:2021

Security Logging
and Monitoring
Failures

A10:2021

Server-Side
Request Forgery

<https://owasp.org/Top10/>

The Egregious Eleven (CSA)

- E1: Data Breaches
- E2: Misconfiguration and Inadequate Change Control
- E3: Lack of Cloud Security Architecture and Strategy
- E4: Insufficient Identity, Credential, Access and Key Management
- E5: Account Hijacking
- E6: Insider Threat
- E7: Insecure Interfaces and APIs
- E8: Weak Control Plane
- E9: Megastructure and Applistructure Failures
- E10: Limited Cloud Usage Visibility
- E11: Abuse and Nefarious Use of Cloud Services



Pitfall #9: Compliance security trainings

- Company-wide “security trainings” for developers
- Hours of slides and videos with a final quiz
- Language specific trainings? Great
 - Just slides?
- We develop in Go but we follow a yearly Java security training
 - ????????



Hands-on to rock it

- Identify people interested in security inside your team
- Create a Security Champions Program
 - Security Champions curriculum
 - Security Champions hands-on trainings
 - Reward
- Organize company-wide CTF (Capture the flag)



Key takeaways: a road to success

- Include security as part of the development process
- Start with one or two tools (SAST and SCA). Tools are not the solution. Your process is
- Adopt a maturity model and stick to it (Only one)
- Integrate vulnerabilities in the developers' tools (Jira, Confluence etc)
- Developers are in charge of defining their pipelines. Do not force ONE security pipeline for everyone
- Involve security only for high-risk change. Do not add the security team as approver for all the MR/PR
- Do continuous Threat Model (15 to 30 min)
- Improve your git security posture (ScoreCards)
- Select custom hand—on trainings for security champions



Questions?