linuxpolska

Beyond SBOM

Securing the Software Supply Chain

*Vulnerability Risk Assessment in **DevSecOps** Environments*

DevSecOps 2023 London

Aleksander Baranowski, Linux Polska
Marek Najmajer, Linux Polska

linuxpolska

Agenda

1. Introduction
2. DevSecOps and Sec Part of it => the Software Composition Analysis – Risk Model
3. Software Life-Cycle and Risks Evolving
4. Whys and Whats of Software Composition Analysis
5. Fundamentals
    - How to qualify and quantify 3rd party software security?
    - Do we have a well established common sense approach to it?
    - So we have at least some empirically tested research-based models?
6. Risks – What if You Don't?
7. Solution – Software Composition Analysis – Risk Model (SCARM)
8. Multi-vector Risk Analysis
    - Contributor profile
    - Project activity/Project dynamics
    - Code quality
    - Vulnerabilities (CVE Dynamics)
9. How to Plug it into the Software Deployment Pipeline

linuxpolska

Marek Najmajer

Product Director, Linux Polska

Academic Teacher, University of Business and Administration in Gdynia, Poland.

Lecturer on Innovative Startups, Business Management and Information Technology.

Aleksander Baranowski

Development Lead, EuroLinux

Open source software security evangelist, digital transformation facilitator - containers, Kubernetes, Ansible, JBoss.

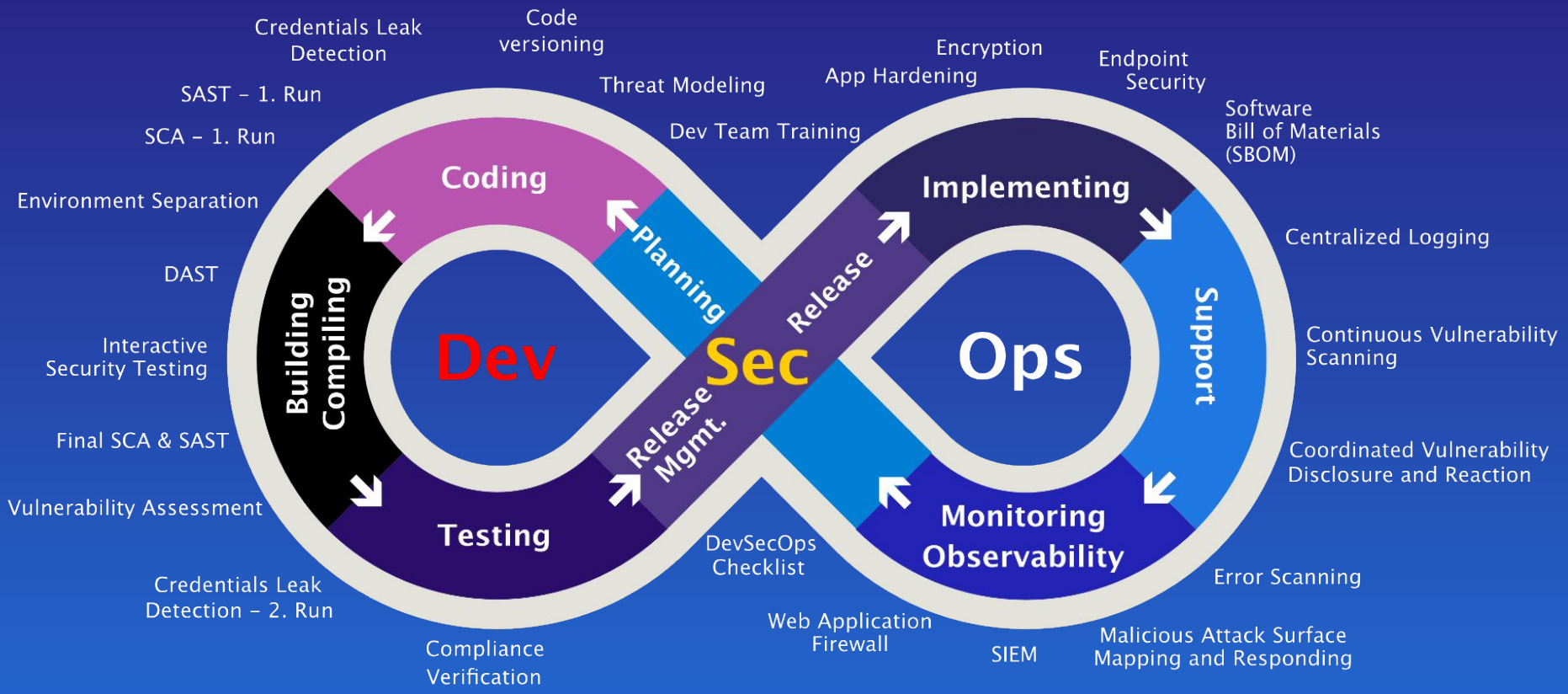
Open source software testing automation promoter.

linuxpolska

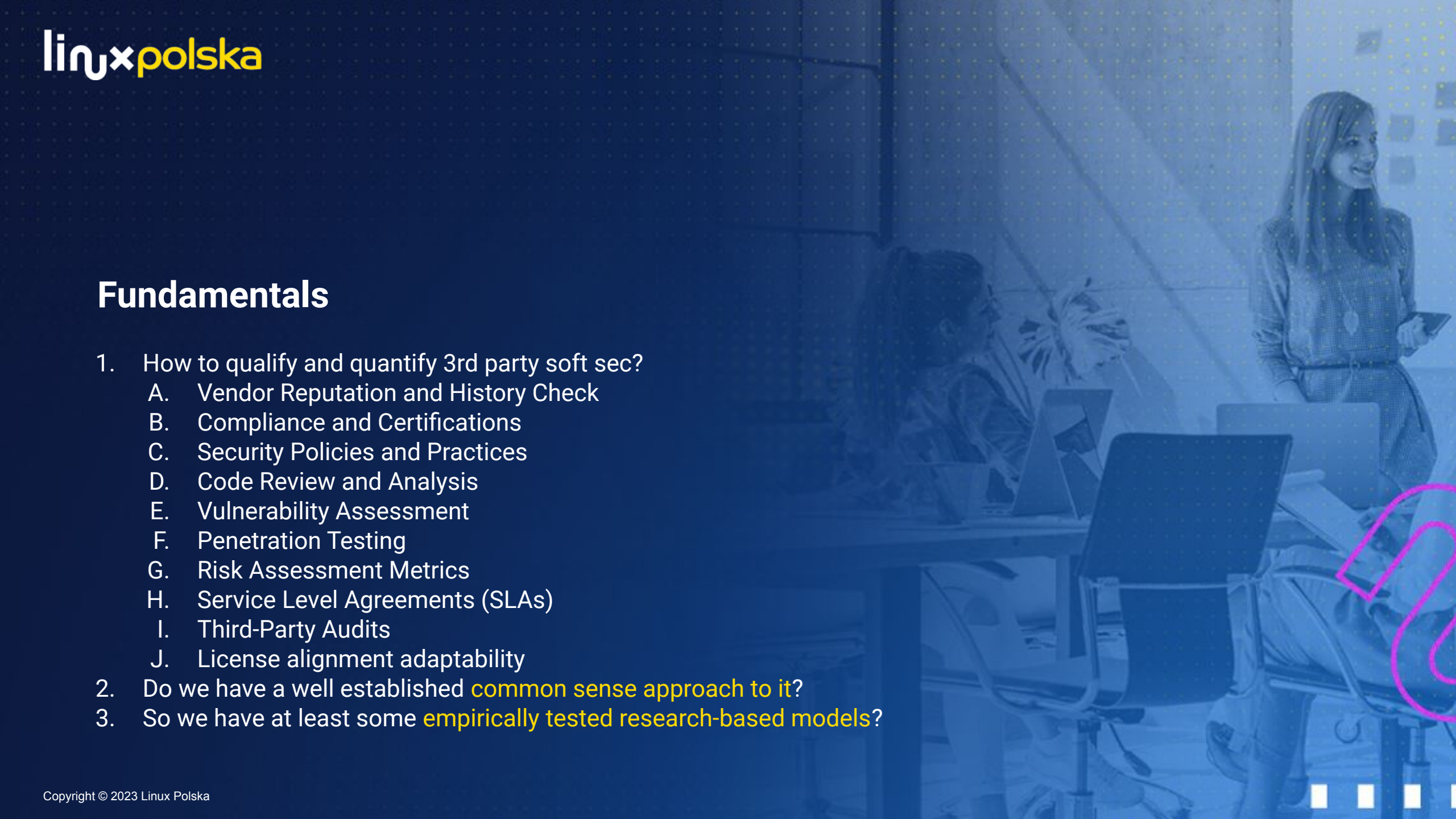
DevSecOps and Sec Part of It => the Software Composition Analysis – Risk Model

- Embracing shift-left approach don’t forget to embed security
- Proactive security with DevOps collaboration
- Implement Security Risk Scoring early in development
- Detect vulnerabilities
- KYCC - know your code and coder
- Beware of mines - know all license FUBARs
- **Shift left, act fast**

linuxpolska



linuxpolska



Fundamentals

1. How to qualify and quantify 3rd party soft sec?
   A. Vendor Reputation and History Check
   B. Compliance and Certifications
   C. Security Policies and Practices
   D. Code Review and Analysis
   E. Vulnerability Assessment
   F. Penetration Testing
   G. Risk Assessment Metrics
   H. Service Level Agreements (SLAs)
   I. Third-Party Audits
   J. License alignment adaptability
2. Do we have a well established common sense approach to it?
3. So we have at least some empirically tested research-based models?

linuxpolska

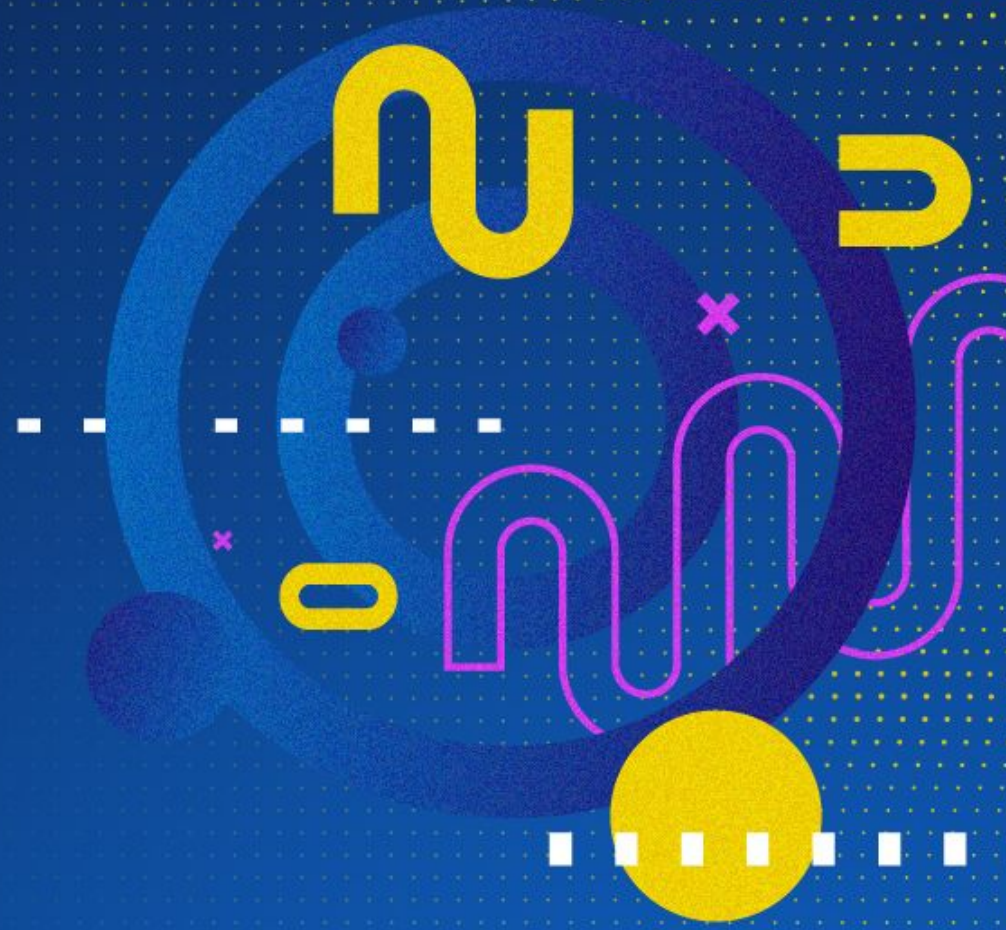
Whys and Whats of Software Composition Analysis

- SCA-like tools identify Third-party Software Components (SBOM)
- Mitigate Security Risks, Ensures Compliance
- Component Risk Scoring Gauges Threat Levels
- Prioritizes Vulnerabilities for Efficient Remediation
- Integrate with Software Dev Pipeline for Proactive Security
- Enhance Reliability and Trust in Software

linuxpolska

Risks – What if You Don’t?

- Increased Vulnerability to Security Breaches
- Potential Legal and Licensing Complications
- Higher Costs from Late-Stage Fixes
- Degraded Software Performance and Reliability
- Eroded Customer Trust and Satisfaction



linuxpolska

Solution:
Software Composition Analysis – Risk Management (SCARM)

- Embrace shift-left approach; embed security
- Proactive security with DevOps collaboration
- Implement Security Risk Scoring early in development
- Detect vulnerabilities
- Shift left, act fast or you be left behind

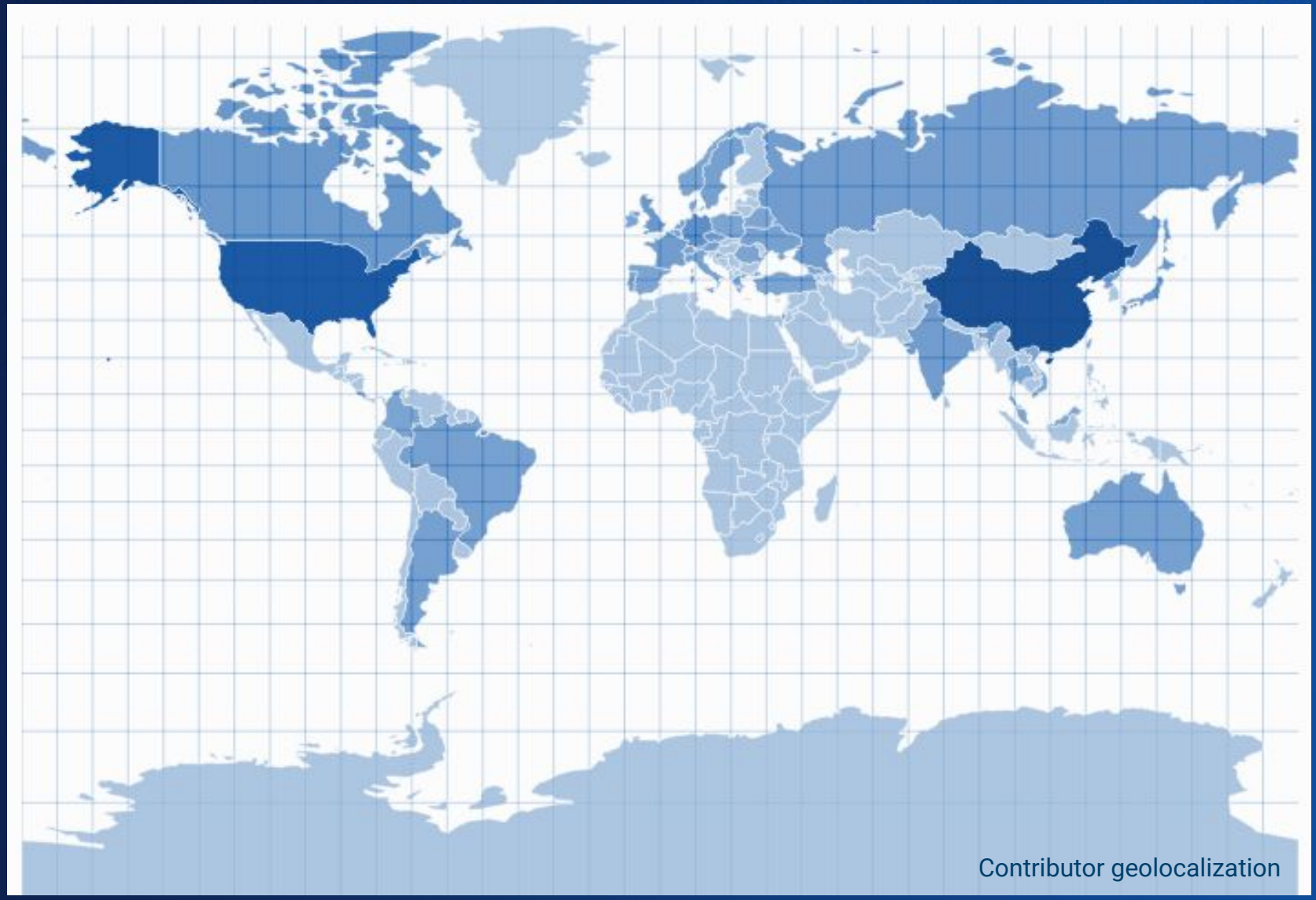
Software Composition Analysis – Risk Management (SCARM)

- Multi-vector analysis of all components (SCA+)
- Current status: 4 vectors/dimensions
  - Software code contributor profile
  - Project dynamics
  - Code quality
  - Vulnerability dynamics (CVE+)

linuxpolska

Contributor Profile

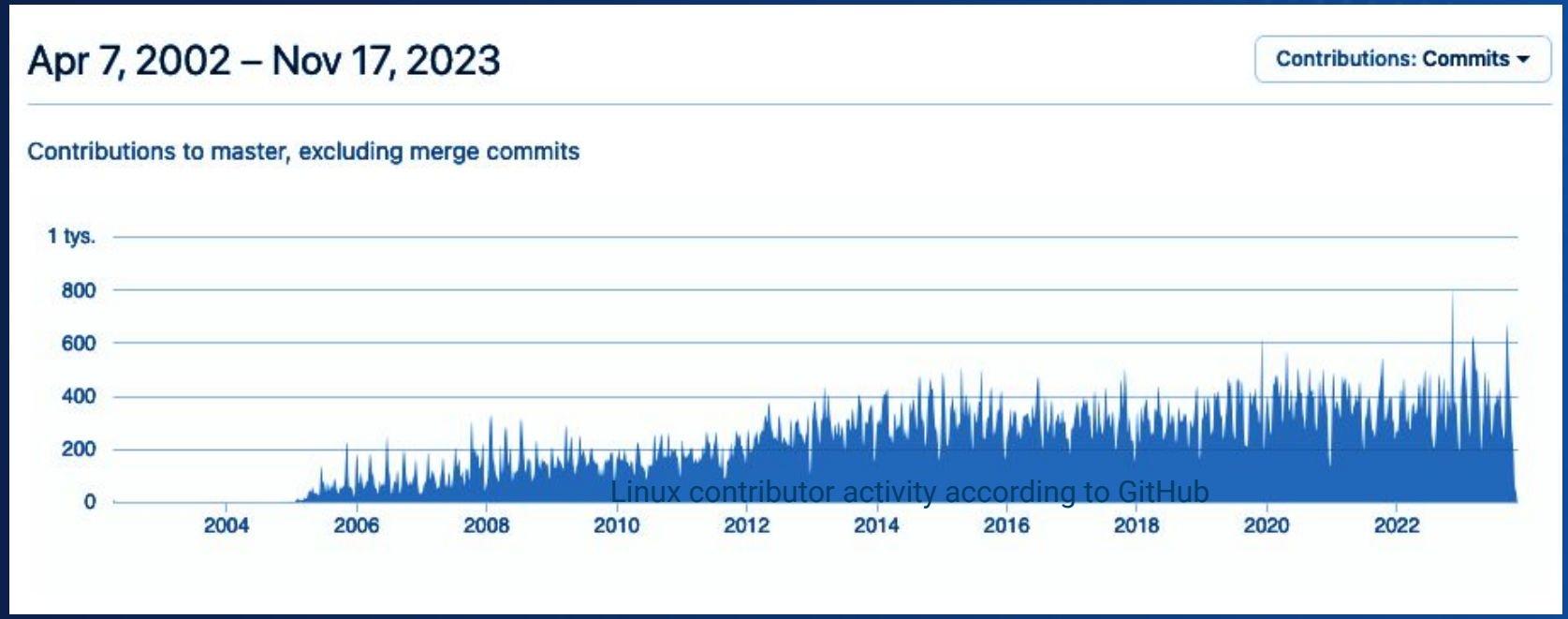
- Project load
- Number of contributors
- Time zone
- Geopolitical risk



Contributor geolocalization

Project Activity = Project Dynamics

- Interest in the project index
- Distribution of activity over time
- Delta of active contributors over time

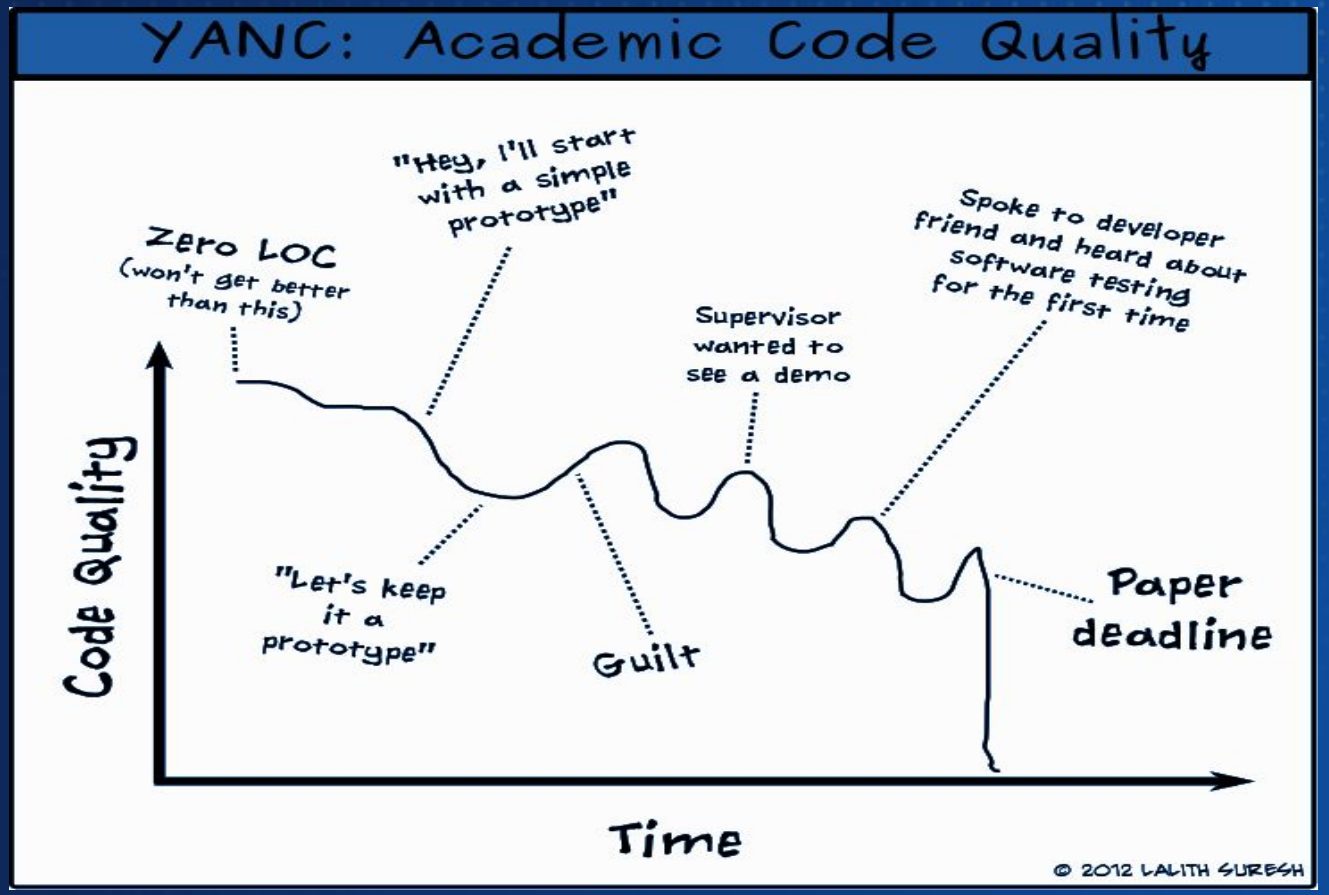


Linux contributor activity according to GitHub

linuxpolska

Code Quality

- Number of lines of code
- Code style
- Sensitive code
- Security
- Performance
- Compatibility
- Code completeness
- Documentation
- Dead code



linuxpolska

Vulnerabilities (CVE Dynamics)

- CVSSv2 score
- CVSSv3 score
- CVSSv4 score (WIP)
- CVE list (last 90 days) with an expand option, there may be many CVEs
- Skating on the thin ice - Pwnie Awards - problem with vendors
- Read Daniel Stenberg (cURL author) - Increased CVE activity in curl?

linuxpolska

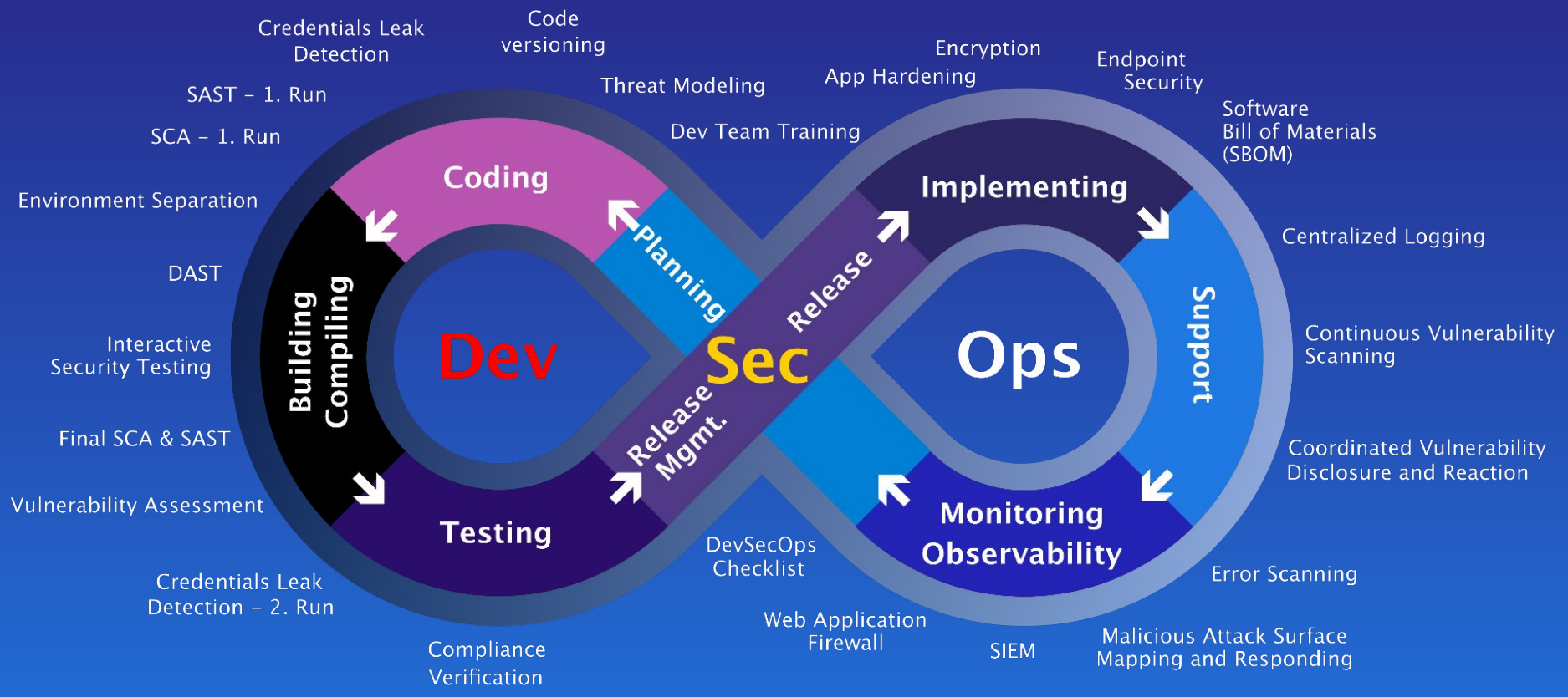
How to Plug it into the Software Deployment Pipeline?

Four touchpoints:

1. Architecture and High Level Design:
    - choosing risky component
2. CI/CD Build and Deployment:
    - spoofed codebase source
3. Maintenance:
    - updating with risky new versions w/o checking
4. Decommissioning of obsolete components:
    - choosing wrong new replacing component

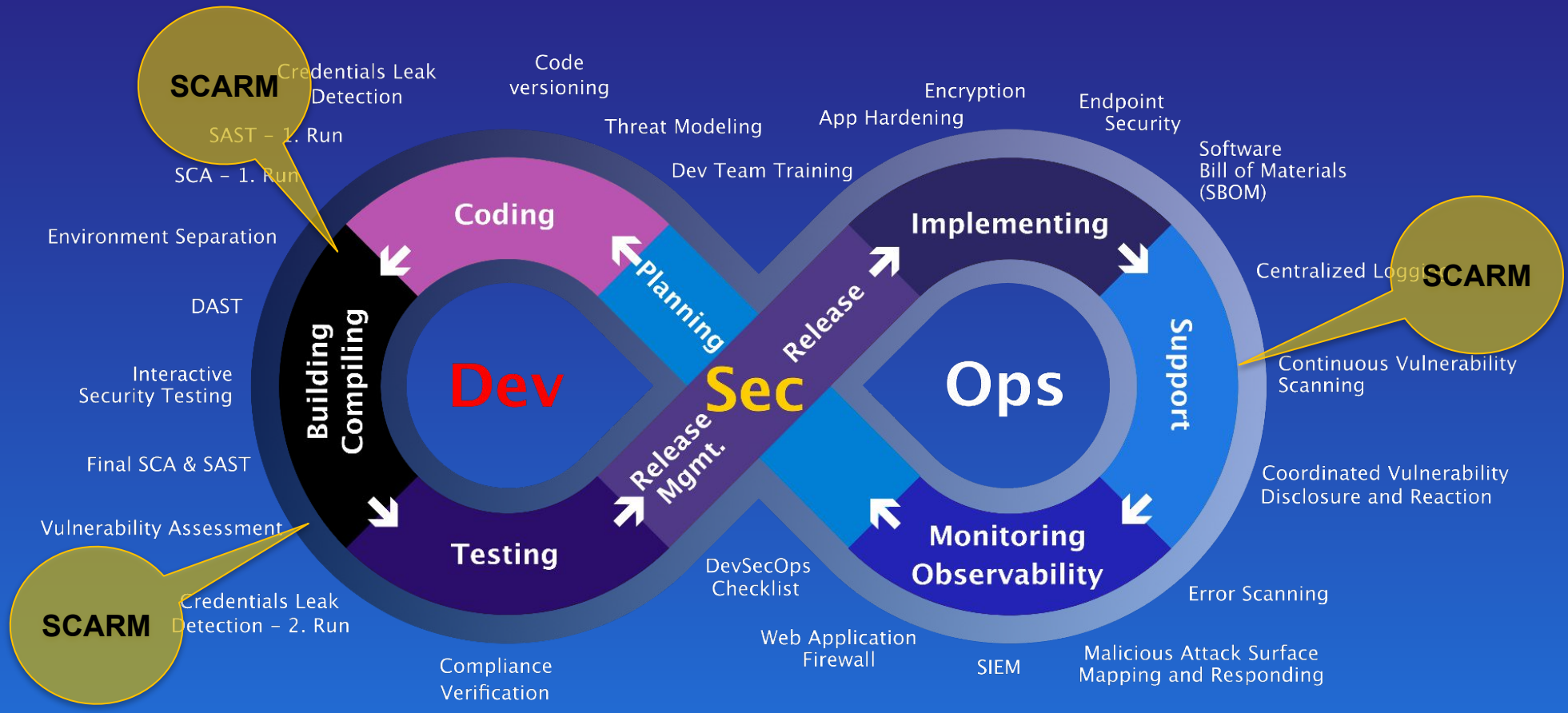
DevSecOps by Linux Polska

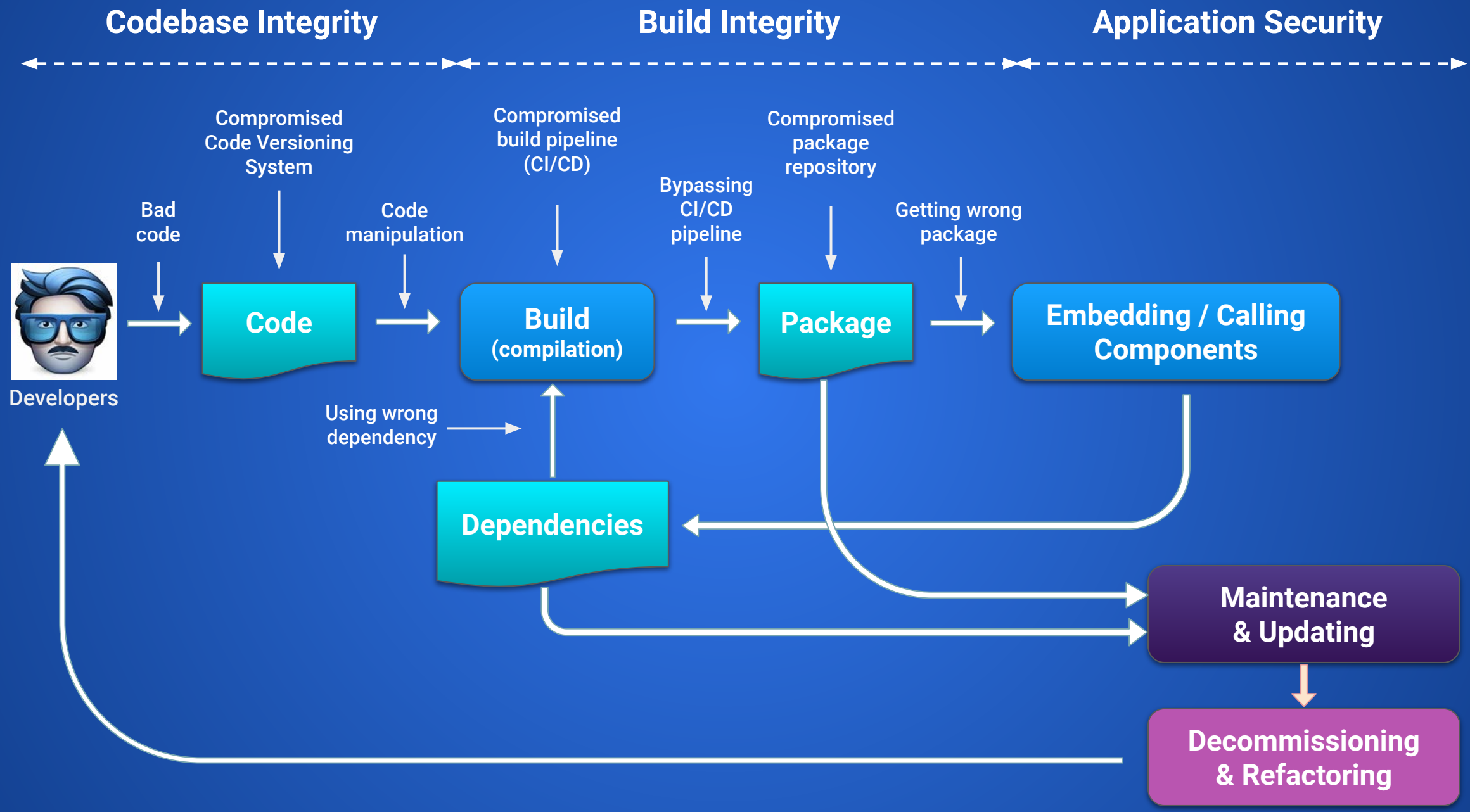
as we see it is **PLANNED** to be

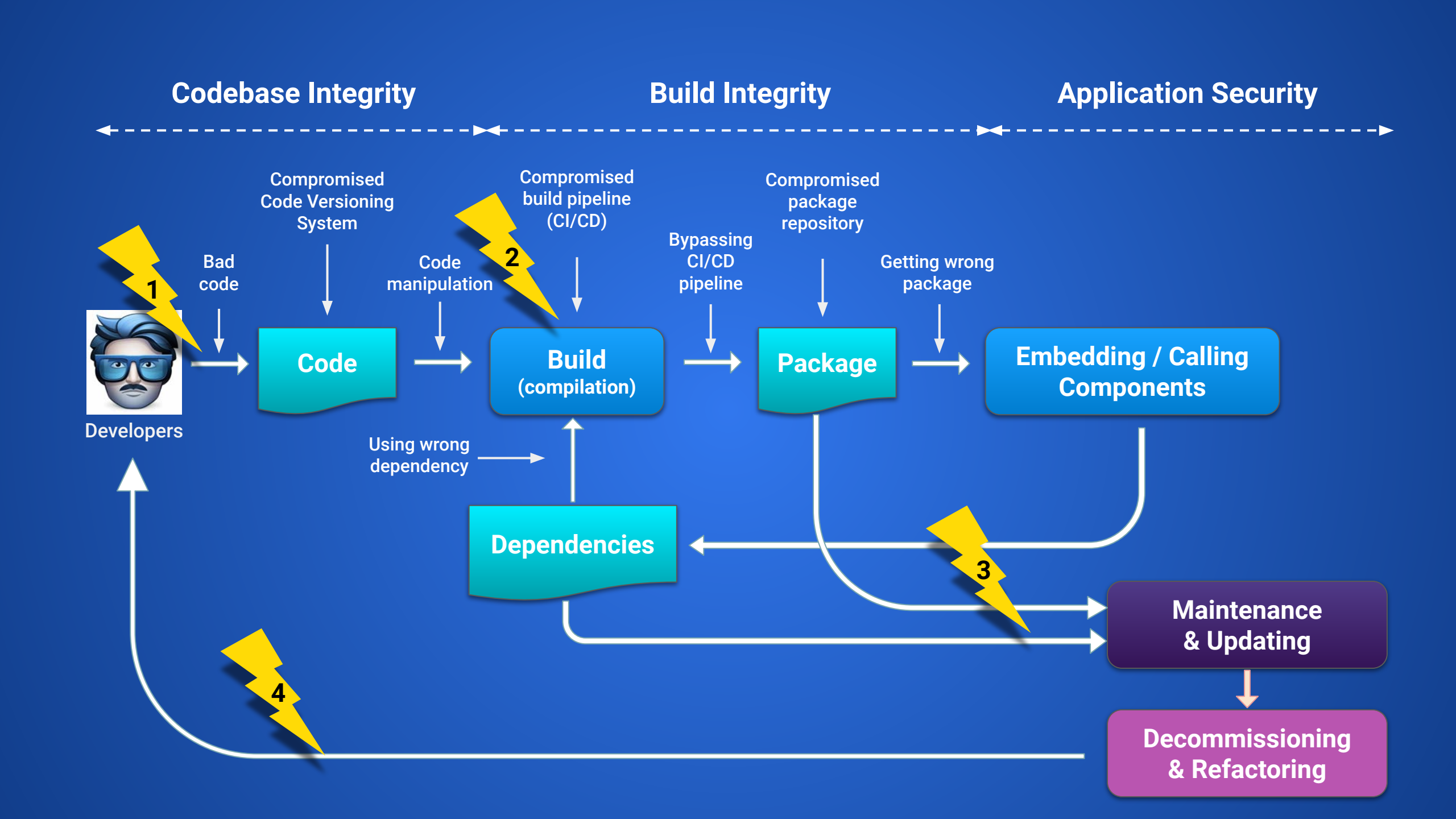


DevSecOps by Linux Polska

as we **MAKE IT WORKS** in the real IT environments




Codebase Integrity
Build Integrity
Application Security
Compromised Code Versioning System
Compromised build pipeline (CI/CD)
Compromised package repository
Bad code
Code manipulation
Bypassing CI/CD pipeline
Getting wrong package
Developers
Code
Build (compilation)
Package
Embedding / Calling Components
Using wrong dependency
Dependencies
Maintenance & Updating
Decommissioning & Refactoring


Codebase Integrity
Build Integrity
Application Security
Compromised Code Versioning System
Compromised build pipeline (CI/CD)
Compromised package repository
Bad code
Code manipulation
Bypassing CI/CD pipeline
Getting wrong package
1
2
Developers
Code
Build (compilation)
Package
Embedding / Calling Components
Using wrong dependency
Dependencies
3
Maintenance & Updating
4
Decommissioning & Refactoring

sourcemation powered by linuxpolska

How to Make It Happens? – Just Start – Here is Some Possibilities Explained

- **Linux Polska R&D Initiative**: funded by the National Center for Research and Development (NCBiR).
- **Open-Source Software Risk Analysis (SCARM)**: focus on identifying and mitigating risks in open-source software.
- **Comprehensive Evaluation System**: assesses software component provenance, version updates, and project activity.
- **SourceMation Portal Integration**: provides access to secure, verified open-source software packages.
- **Security & Stability Assessment:** enables detailed evaluation of open-source software security level and stability.
- **Effective Risk Management**: supports informed decision-making in IT ecosystems.
- **Practical Usage of CVSS Methodology**: utilizes the Common Vulnerability Scoring System for robust risk analysis of open-source packages.

The project "System for analysis of risks present in software packages originating from open-source projects" is co-financed by the European Regional Development Fund under the Operational Program Intelligent Development. The aim of the project is to design and develop a prototype system for the production and distribution of software coming from open source projects while meeting the security and risk management requirements of mission-critical systems.

Fundusze Europejskie Inteligentny Rozwój

Rzeczpospolita Polska

NCBR Narodowe Centrum Badań i Rozwoju

Unia Europejska Europejski Fundusz Rozwoju Regionalnego

sourcemation powered by linuxpolska

New Web Service Platform Helping DevSecOps Teams to Identify Potential Risks of Open Source Utilizing

- SourceMation is a web service containing libraries of software and applications available for automated deployment
- SourceMation's key element - open software software security assessment
- SourceMation helps identify potential risks of open source software utilize through an:
    - in-depth analysis of code quality
    - possible software vulnerabilities
    - its history
    - and many other factors

  to assess the risks associated with a specific open source project

sourcemation powered by linuxpolska

Streamline Your Security Assessments with Our Detailed Information and Comprehensive Risk Scoring System

- Open source components risks in detail
- Easy to read, easy to compare
- Comprehensive codebase risk scoring on:
  - component project activity
  - geopolitical risk linked with contributor provenance
  - project interest rate
  - code size, style, completeness, dead code
  - code sensitivity, security, performance compatibility
  - project documentation
  - CVSS

Constantly expanding list of software

sourcemation powered by linuxpolska

see you soon at sourcemation.com

linuxpolska

Thank you very much for your attention
We invite you to contact us:

Aleksander Baranowski, @make_void
Marek Najmajer, @marqoz