



How to Assess OSS Project Health

A Holistic View of Open Source Risk

Daniel Izquierdo <dizquierdo@bitergia.com>
National Open Source Innovation Summit, 2025

Meet **Daniel Izquierdo**

Co-founder of Bitergia

- CEO @ Bitergia
- President @ InnerSource Commons
- Governing Board @ CHAOSS
- Governing Board @ Apereo Foundation



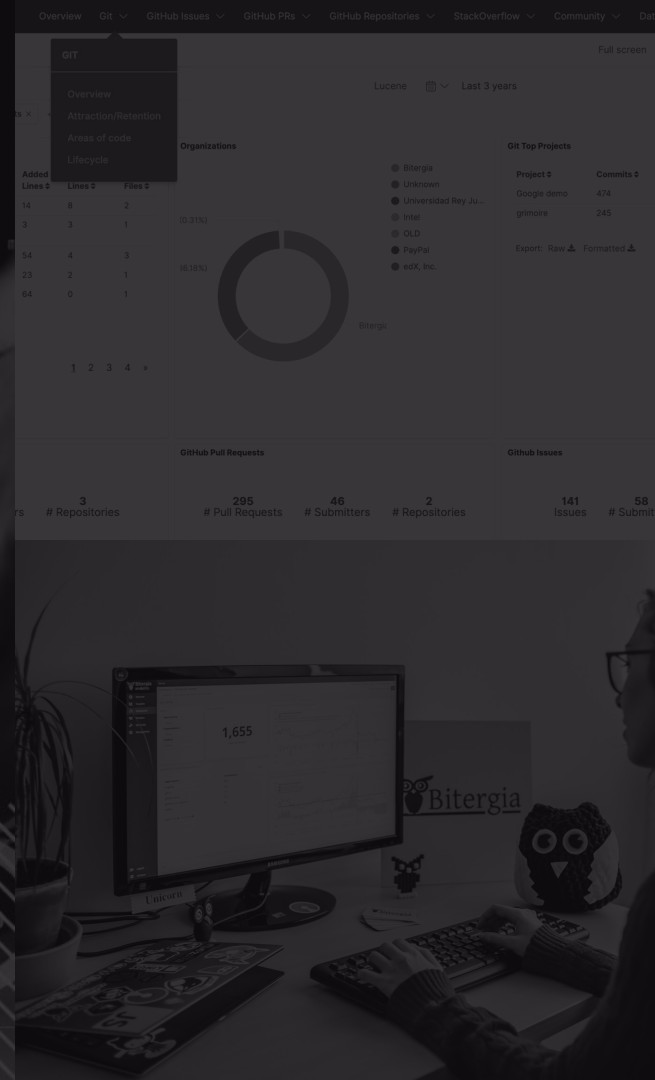
dizquierdo@bitergia.com

linkedin.com/in/dicortazar

PhD in empirical software engineering



Context



We're coming from a context where **code** is treated as **literature** and moving into a context where you are **liable** of the code you produce **if** it is **commercially available**

Liability means that you can be sued for damages



There are three new regulations
in place in the EU on

- Liability - PLD
- Cybersecurity - EU CRA
- AI - AI Act

Law is more important than OSS
licenses



15. Disclaimer of Warranty.

THERE IS **NO WARRANTY** FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES **PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND**, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. **SHOULD THE PROGRAM PROVE DEFECTIVE**, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

<https://www.gnu.org/licenses/gpl-3.0.en.html#license-text>



16. *Limitation of Liability.*

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

<https://www.gnu.org/licenses/gpl-3.0.en.html#license-text>



Now if you want to publish OSS
code for commercial purposes
you have to comply with the EU
CRA and the AI Act



Home / News / [Webinars on recent EU regulation for non-EU audiences](#)

Webinars on recent EU regulation for non-EU audiences

07 AUGUST 2024

Author: Ciaran

<https://openforumeurope.org/webinars-on-cra-ai-pld/>



Ciarán O'Riordan

SENIOR POLICY ADVISOR

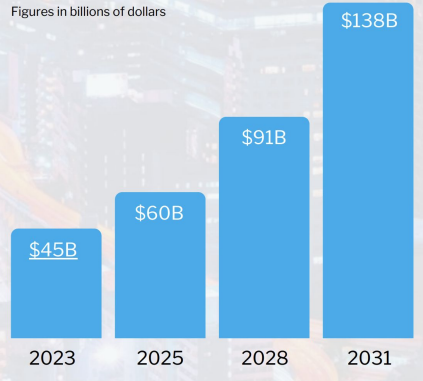


Why this matters to me?

DAMAGE COSTS

Cybersecurity Ventures predicts that the global cost of software supply chain attacks to businesses will reach nearly \$138 billion by 2031, up from \$60 billion in 2025, based on 15 percent year-over-year growth.

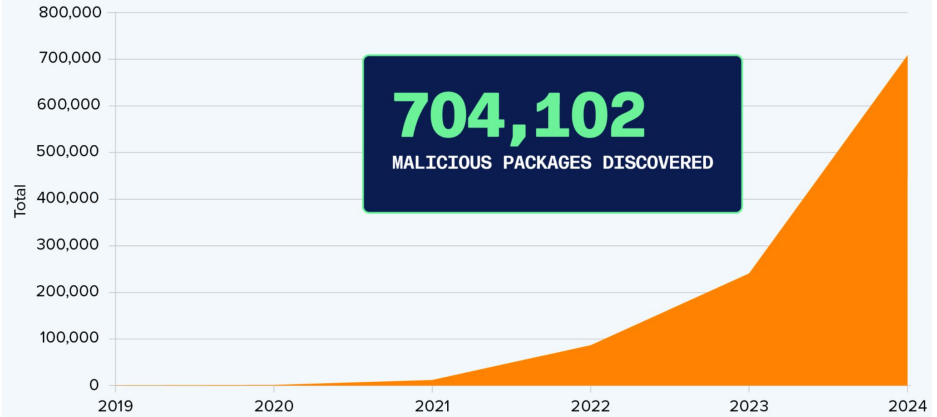
Figures in billions of dollars



<https://go.snyk.io/2023-supply-chain-attacks-report.html>

FIGURE 1.1

Next Generation Software Supply Chain Attacks (2019-2024)



Malicious OSS packages discovered (2019-2024).

<https://www.sonatype.com/state-of-the-software-supply-chain>



A Summary of Census II: Open Source Software Application Libraries the World Depends On

JASON PERLOW | 07 MARCH 2022



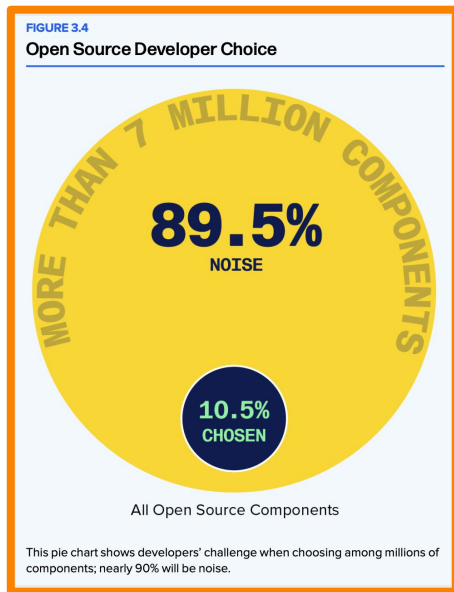
Introduction

It has been estimated that Free and Open Source Software (FOSS) constitutes 70-90% of any given piece of modern software solutions. FOSS is an increasingly vital resource in nearly all industries, public and private sectors, among tech and non-tech companies alike. Therefore, ensuring the health and security of FOSS is critical to the future of nearly all industries in the modern economy.

<https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on>



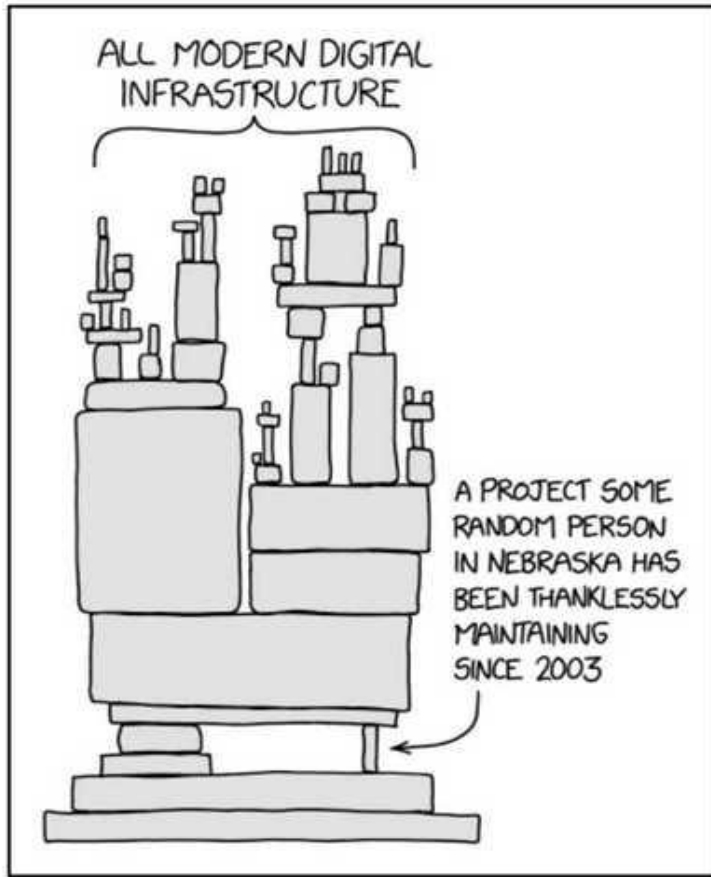
Dilemma: Choice and Maintenance



<https://www.sonatype.com/state-of-the-software-supply-chain>

Sonatype 9th State of Software Supply Chain report:
“Consider this: last year, we revealed that a staggering **85% of projects in Maven Central** — the largest public repository for Java open source components — **are inactive**. In other words, developers are faced with a perplexing array of choices, with only a fraction of them leading to active, well-maintained projects.”





How do I know
the unknown?



SBOMs and Trust



Software ages like **Milk, not Wine**

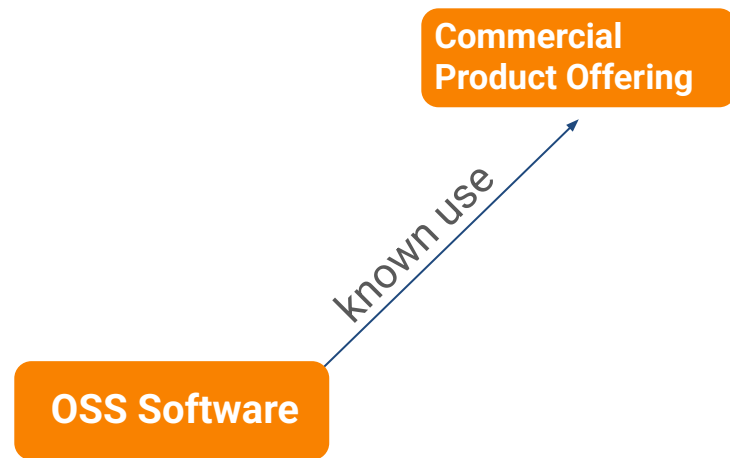


Trust: Expiration Label and Source Information



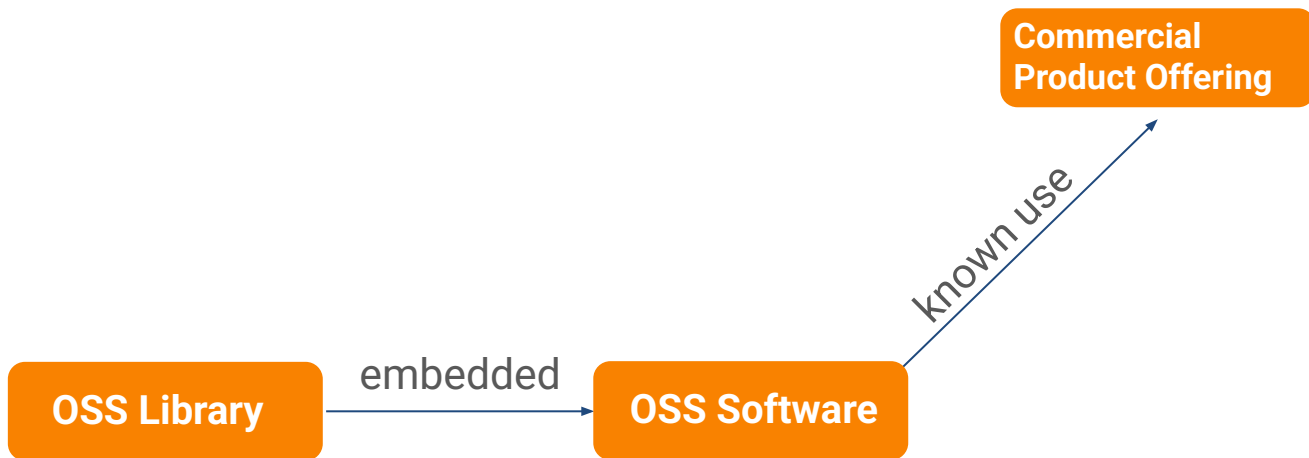
Context: **Unmanaged OSS Use** → **Unknown Risk**

- Developers use open source software
- 70% - 95% of software includes open source



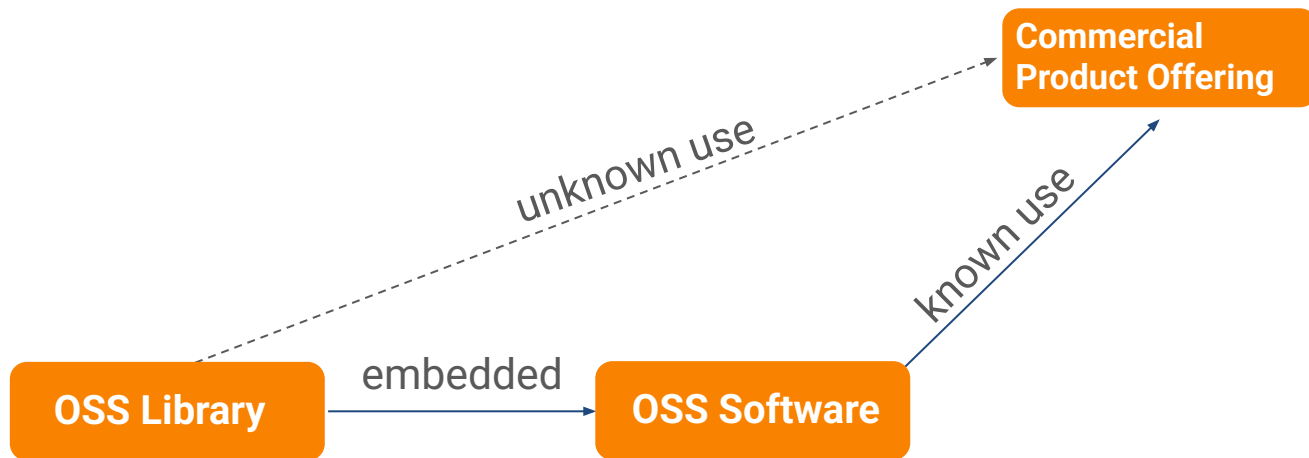
Context: **Unmanaged OSS Use** → **Unknown Risk**

- Developers use open source software
- 70% - 95% of software includes open source
- Unmanaged OSS use → unknown dependencies



Context: Unmanaged OSS Use → Unknown Risk

- Developers use open source software
- 70% - 95% of software includes open source
- Unmanaged OSS use → unknown dependencies → **unknown risk**



Problem: Trust in OSS Libraries to Manage Risk

180

average number of components
per application | **EVEN SMALL APPLICATIONS
FACE UNMANAGEABLE WORKLOADS**

<https://www.sonatype.com/state-of-the-software-supply-chain>



Problem: Trust in OSS Libraries to Manage Risk

Licenses

Vulnerabilities

Under-maintained projects

180

average number of components
per application | **EVEN SMALL APPLICATIONS**
FACE UNMANAGEABLE WORKLOADS

<https://www.sonatype.com/state-of-the-software-supply-chain>



Problem: Trust in OSS Libraries to Manage Risk

Licenses

- License scanners

Vulnerabilities

- Software Composition Analysis (SCA)
- Vulnerability Databases

Under-maintained projects

- Community Health Metrics (CHAOS)
- Pay for support

180

average number of components
per application | **EVEN SMALL APPLICATIONS**
FACE UNMANAGEABLE WORKLOADS

<https://www.sonatype.com/state-of-the-software-supply-chain>

Missing: Forward looking risk



Example of Kubernetes' Go Dependencies

Indicators for Risk: “Under-maintained Projects”

“Community Smells” include 7 metrics:

Community cannot handle **workload**

- Backlog Management Index
- Review Efficiency Index

Community does not address **work quickly**

- Median Lead Time for Issues
- Median Lead Time for Pull Requests

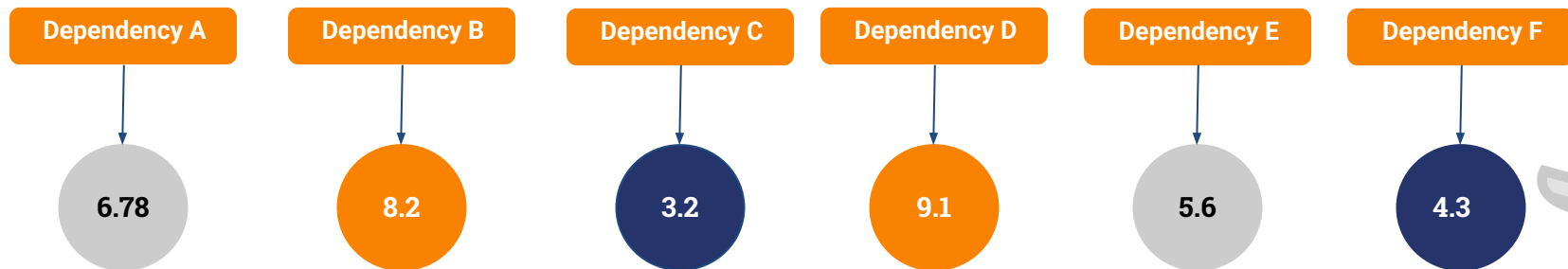
Community lacks sufficient **talent**

- Retention Rate
- Growth of Active Contributors
- Contributor Absence Factor (aka Bus or Pony Factor)

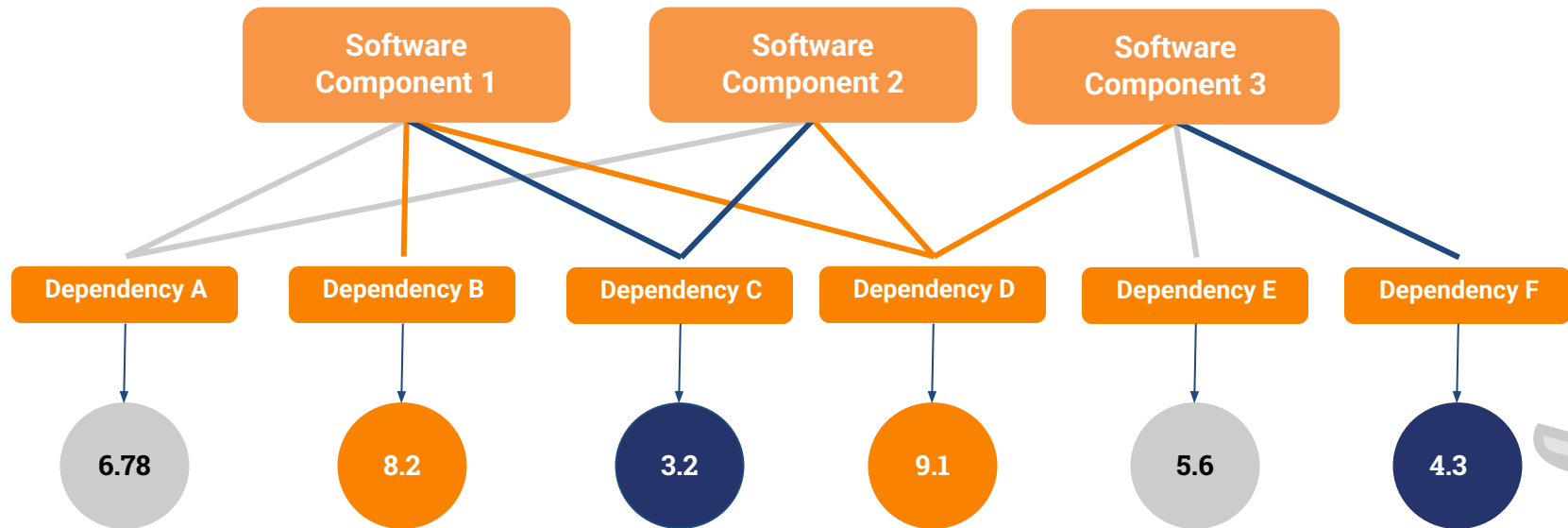


A single Risk Score per OSS library

7 metrics, normalized, and
combined into one score for each dependency



Risk Model - Aggregate By Component



Risk Model Overview Dashboard

Check the [Risk Model Help Dashboard](#) for more information about this analysis.

For more details, pin a filter by origin and visit the [Risk Model Dashboard for Individual Projects](#).

Filters

Team

Select...

Project Category

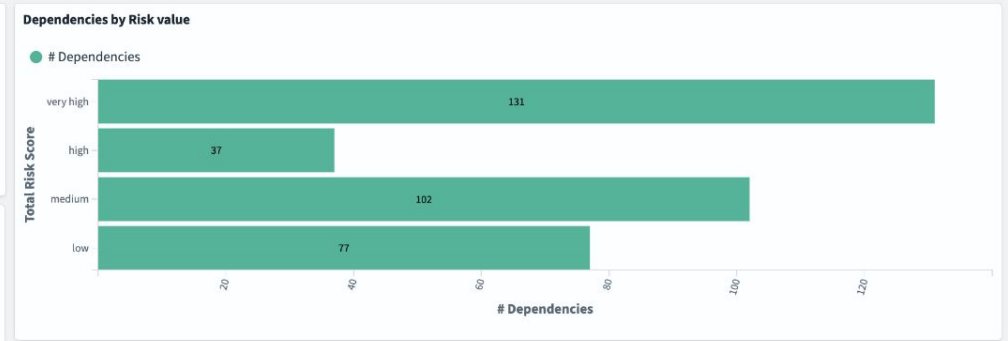
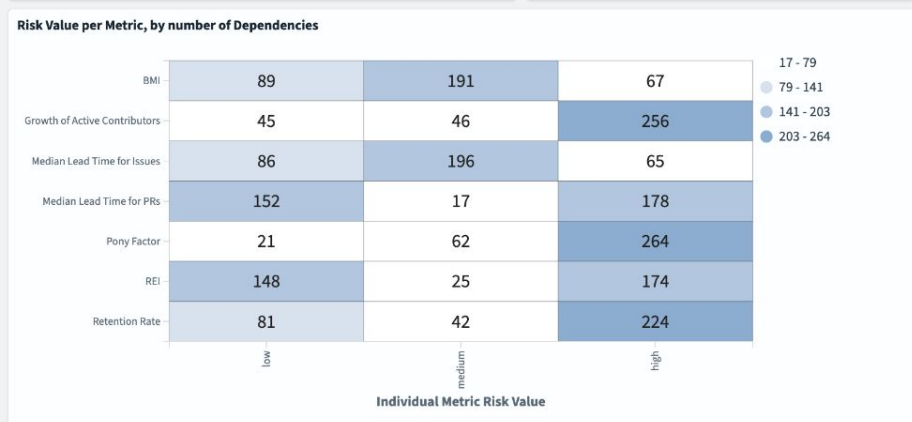
Kubernetes - Golang Deps ×

Overview


347

Dependencies analyzed

Bitergia Team



Overall results by dependency repository

Filter... 

Repository ↕	Category ↕	Risk Value ↕	Risk Score (over 10) ↕	# "Low risk" metrics ↕	# "Medium risk" metrics ↕	# "High risk" metrics ↕	Last analyzed on ↕
https://github.com/json-iterator/go	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/spfl3/afero	Kubernetes - Golang Deps	very high	10	0	0	7	Jun 27, 2024 @ 12:35
https://github.com/Azure/go-ansiterm	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/Thalesignite/crypto11	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/coreos/go-semver	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/curioswitch/go-reassign	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/davecgh/go-spew	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35

Export: [Raw](#) [Formatted](#)

1

2

3

4

5

...

27

Risk Model Overview

Check the Risk Model Help

For more details, pin a filter

Filters

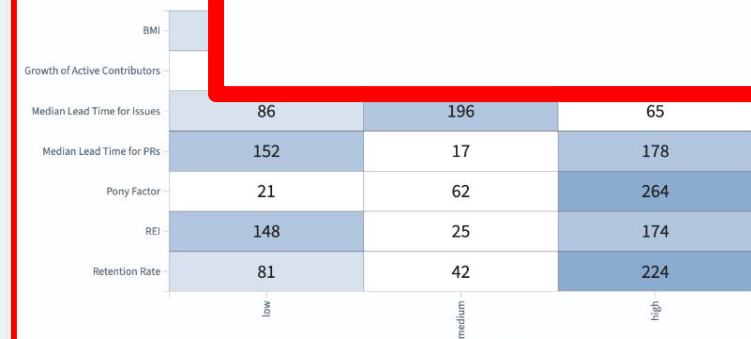
Team

Select...

Project Category

Kubernetes - Golang D

Risk Value per Metric, by number of Dependencies



Risk Value per Metric, by number of Dependencies

BMI	89	191	67	17 - 79
Growth of Active Contributors	45	46	256	79 - 141
Median Lead Time for Issues	86	196	65	141 - 203
Median Lead Time for PRs	152	17	178	203 - 264
Pony Factor	21	62	264	
REI	148	25	174	
Retention Rate	81	42	224	
	low	medium	high	

Individual Metric Risk Value

	Deps						12:35
https://github.com/coreos/go-semver	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/curioswitch/go-reassign	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
https://github.com/davecgh/go-snew	Kubernetes - Golang Deps	very high	9.29	0	1	6	Jun 27, 2024 @ 12:35
Export: Raw 📄 Formatted 📄							

Experiences and Analysis to share

Digital Sovereignty

Merge & Acquisition, due diligence

Social engineering attacks (e.g., XZ)

Code review fairness

Competitive analysis

Footprint analysis / developers migrations

Influence indicators



Experiences and Analysis to share

Digital Sovereignty

Merge & Acquisition, due diligence

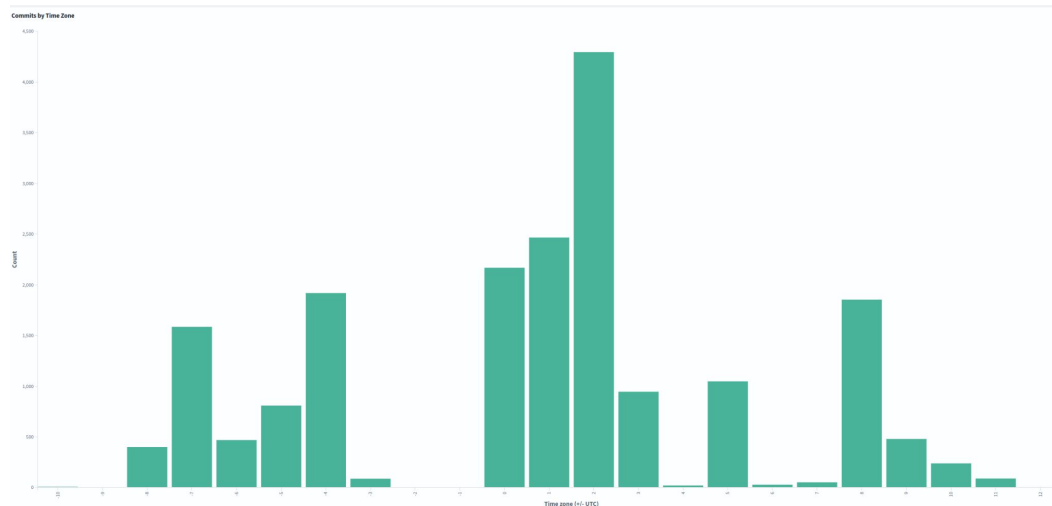
Social engineering attacks (e.g., XZ)

Code review fairness

Competitive analysis

Footprint analysis / developers migrations

Influence indicators



Experiences and Analysis to share

Digital Sovereignty

Merge & Acquisition, due diligence

Social engineering attacks (e.g., XZ)

Code review fairness

Competitive analysis

Footprint analysis / developers migrations

Influence indicators

Re: [xz-devel] XZ for Java

Lasse Collin | Wed, 08 Jun 2022 03:28:08 -0700

On 2022-06-07 Jigar Kumar wrote:

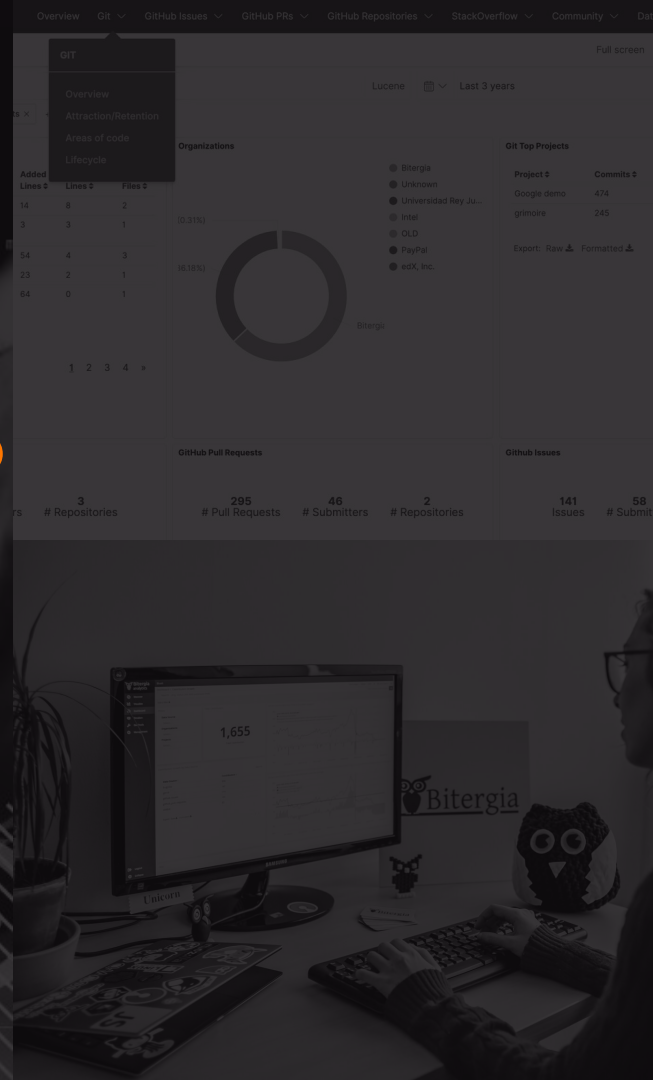
> Progress will not happen until there is new maintainer. XZ for C has
> sparse commit log too. Dennis you are better off waiting until new
> maintainer happens or fork yourself. Submitting patches here has no
> purpose these days. The current maintainer lost interest or doesn't
> care to maintain anymore. It is sad to see for a repo like this.

I haven't lost interest but my ability to care has been fairly limited mostly due to longterm mental health issues but also due to some other things. Recently I've worked off-list a bit with Jia Tan on XZ Utils and perhaps he will have a bigger role in the future, we'll see.

It's also good to keep in mind that this is an unpaid hobby project.

Anyway, I assure you that I know far too well about the problem that not much progress has been made. The thought of finding new maintainers has existed for a long time too as the current situation is obviously bad and sad for the project.

Where to start?



Good News

Everything is open by default in open source

You can make decisions based on real data

Projects as Kubernetes are backed by the industry

There are companies specialized in helping in this journey

And open source foundations are enforcing good engineering and security practices

Modern development practices come from open source



Getting Started

Prepare for the law, including your suppliers

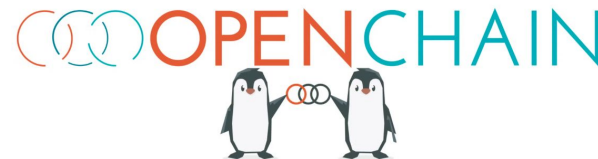
Discover through SBoM analysis unmaintained projects

Be part of the conversation with your critical OSS projects

Enforce good software engineering and security practices

Hire open source expertise

This should be part of the usual internal risk management processes



BUILDING TRUST IN THE SUPPLY CHAIN SINCE 2016





Luis Cañas-Díaz



Georg Link, PhD

Shining Light on the Open Source Supply Chain: **The Risk in Community Health**

Ubuntu Summit, Den Haag, Netherlands, Oct. 25, 2024



OPEN CODE EXPERIENCE

Reducing risk in software supply chains: A project health perspective with a Kubernetes example

Peter Eichelsheim, ING
Georg Link, Bitergia



The Role of **Open Source Management** Talent in Ensuring Software Ecosystem Stability

オープンソース人材の採用とソフトウェアエコシステムの安定性

Ana Jiménez, Linux Foundation
Daniel Izquierdo, Bitergia



Defining the Limits of Risk

OSS Summit EU 2022

Daniel Izquierdo Cortázar

Software
Development
Analytics



How to Assess OSS Project Health

A Holistic View of Open Source Risk

Daniel Izquierdo <dizquierdo@bitergia.com>
National Open Source Innovation Summit, 2025