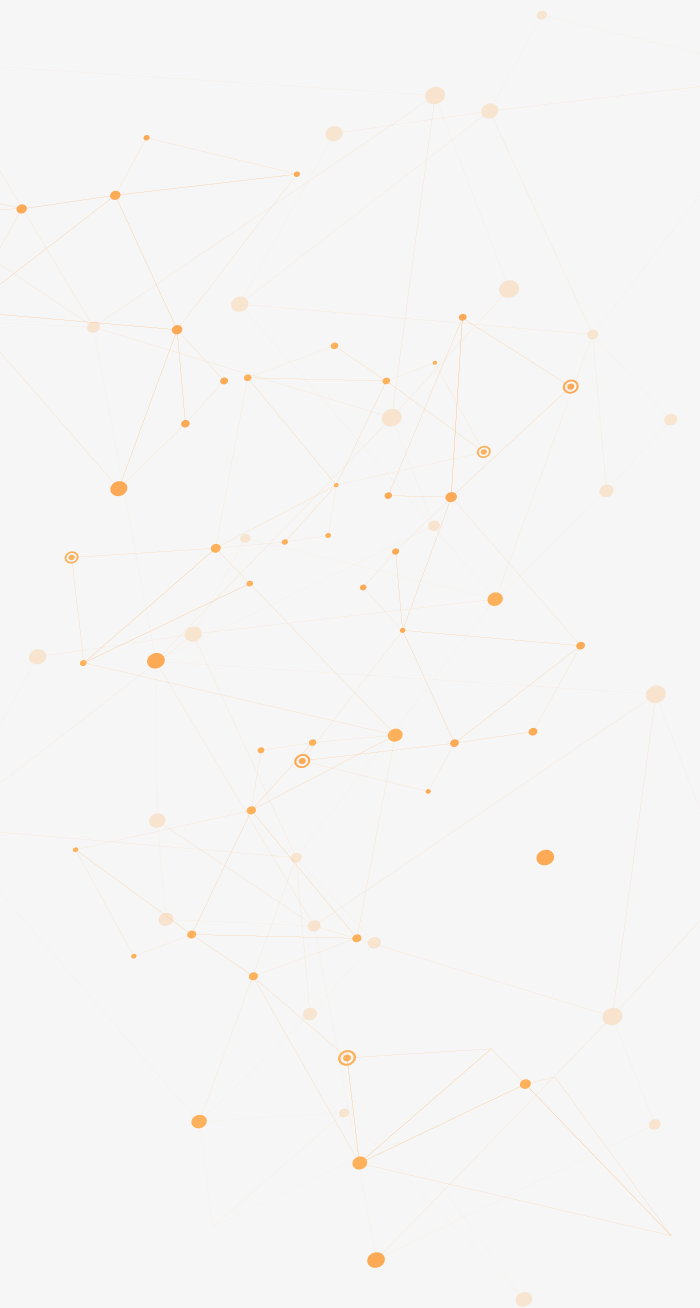
Bitergia

Bitergia Radar Whitepaper

PROJECT HEALTH IS THE RISK FACTOR FLYING UNDER THE RADAR FOR SECURING SOFTWARE SUPPLY CHAINS AND ENSURING SOFTWARE RESILIENCE

INDEX

Executive Summary



The landscape of software development has fundamentally shifted. Open source components now form the backbone of the vast majority of software supply chains. Security threats and government regulations to secure software supply chains are both on the rise. That's why it is more essential than ever to be able to trust the dependencies that are the foundation of your software. **Trust comes from knowing that dependencies are properly maintained and not open to vulnerabilities or attacks.** While security scanning and license compliance have traditionally been the primary pillars of open source strategy, the increasing complexity and vulnerability of open source software supply chains necessitate a third, crucial pillar: project health.

This whitepaper explores the critical importance of project health in achieving software resilience, navigating evolving regulations, and mitigating the rising risks of an increasingly complex software world. **It also presents Bitergia Radar–a robust solution for assessing and improving project health.** The Bitegia Radar Services provide organizations with the insights they need to proactively address risk in their software supply chains, save money, and have trust that their software foundation is strong.

The Open Source Landscape Is Changing Fast

Open Source Is Everywhere

Open Source Software (OSS) has become the indispensable engine of modern technology, powering nearly every digital innovation. In fact, the 2024 Open Source Security and Risk Analysis Report found that **96% of the total code base is open source**. This widespread reliance on OSS offers undeniable advantages, including rapid development cycles, cost-effectiveness, and access to a vast pool of collaborative expertise.

However, this reliance also presents significant security and maintainability risks. In the current open source context, organizations are responsible for a complex supply chain and they need ways to assess the riskiness and sustainability of their dependencies. They need ways to be able to trust their software dependencies.

Threats to Open Source Are Exploding

As adoption of open source has risen, so have costly attacks to that software. Gartner anticipated that **45% of organizations will have experienced a software supply chain attack by 2025**. A report by Sonatype found "a massive year-over-year increase in cyberattacks aimed at open source project ecosystems." By massive, they were referring to a "**700% jump in repository attacks** over the last three years." High-profile incidents, such as the Log4j vulnerability, have underscored the potential for widespread disruption and damage.

Supply chain attacks can be devastating. Research into the SolarWinds breach, for example, found that it cost businesses a stunning average amount of **11% of their annual revenue**. In addition to the financial impact, there's the disruption to products and processes as teams scramble to fix the problem.

These attacks increased even as investments in third-party cybersecurity risk management (TPCRM) increased. That's why incorporating project health as a core pillar of your open source strategy is important. It is the missing piece to being able to fully trust your dependencies, and building software resilience. If a dependency is not maintained, for example, or is maintained by only one or two people, it is left more open to attacks. Assessing the health of dependencies is a solution that helps organizations get ahead of risks. With data, organizations can identify troublesome dependencies and invest in ones that will last so that they can trust their dependencies.

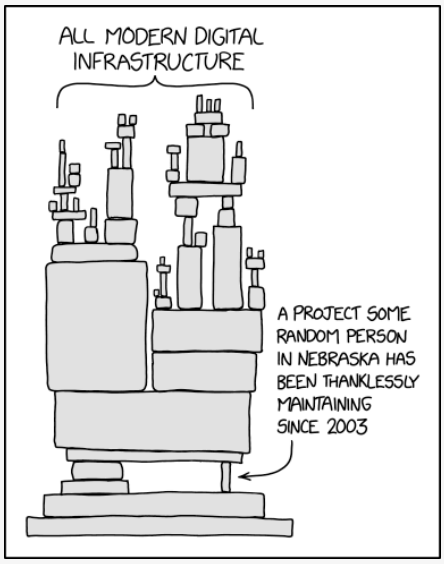
Software Is Becoming a Regulated Industry

Meanwhile, regulatory landscapes are also changing quickly. The US Government, with Executive Order 14028 and related legislation, has been increasing requirements for cybersecurity, but other regions are also increasing requirements. The European Cyber Resilience Act (CRA) is a landmark piece of legislation that places significant responsibility on manufacturers and distributors to ensure the cybersecurity of their products, including OSS components. **The CRA mandates that organizations must "exercise due diligence" in managing their software supply chain,** requiring proactive monitoring, risk assessment, and mitigation strategies. This shift in regulatory focus underscores the urgent need for organizations to adopt a comprehensive approach to OSS security, risk assessment, and software resilience.

There's a Need for More Robust Risk Assessment for Open Source Dependencies

Software supply chains are often on shaky ground, relying on under-maintained open source projects that pose risk. This is especially problematic in a software environment where threats are on the rise. A popular xkcd cartoon depicts this point. In it, a tower of haphazard blocks–labeled "All modern digital infrastructure"–teeters on top of one single block (or "A project some random person in Nebraska has been thanklessly maintaining since 2003").

It's funny, and also a little scary.



In any given company, the software "tower" is built with an array of building blocks from OSS dependencies. In fact, the 2024 Open Source Security and Risk Analysis Report found that 96% of the total code base is open source.

Perhaps that tower is strong, or perhaps there are project health issues that could send it tumbling. **Knowledge is the key to managing risk and achieving trust in your dependencies.**

Companies build their software using OSS dependencies as much of its base. These blocks are powerful resources for innovation and faster time to market, but it's important to know when they're not sustainable. According to an IEEE study, "**systems using outdated dependencies are four times as likely to have security issues** as opposed to systems that are up-to-date." This stark number points to the urgency of understanding the maintenance activity of OSS dependencies.

In the context of all the many changes in the software development landscape, understanding and managing the health of OSS projects is no longer a mere best practice but a critical necessity to ensure security, sustainability, and project resilience.

Bitergia Radar Services help organizations rise to this moment. These services harness project health metrics and insights to empower organizations to proactively address risks and to help ensure trust in a sustainable software supply chain. We aim to bring trust that the open source software "tower" you've built won't go toppling.

Project Health is Key to Trusting Your Dependencies

Can You Trust Your Dependencies?

Can you trust your dependencies? The short answer is: no, not without insights about project health. Trust comes from a complete understanding, from approaching your software resilience with eyes open.

There are many benefits to prioritizing project health:

- **Proactively identifying risks:** Spot potential problems before they impact your project.
- **Making informed decisions:** Choose open source components that are not only secure and compliant but also healthy and sustainable.
- **Reducing risk:** Avoid the hidden costs associated with unhealthy projects.
- **Building more resilient software:** Rely on projects that can withstand the test of time.

What Is Project Health?

Think of a responsible open source strategy as three columns holding up a massive structure. The structure in this metaphor is an organization's software. Most companies focus on two of the three pillars: security and compliance. But **project health is the third essential pillar**. It adds the stability and software resilience to withstand increasing pressures. Understanding project health is essential to having trust in your dependencies and avoiding costly attacks.

Project health is the overall well-being and resilience of a software development project from a sociotechnical point of view. By sociotechnical, we're referring to the contributor community and their development activity. It is the intersection of people (maintainers) and processes (activity) in a technical environment. OSS resilience is the capacity of an OSS project to recover quickly from difficulties and continue releasing quality software. A healthy project–one that has a thriving community and efficient maintainer activity–is resistant to attacks and sustainable over the long term. On the other hand, an unhealthy project poses risk. And organizations have a responsibility to know that risk and to act on it.

Why Do Project Health Metrics Matter?

We are experts at measuring project health and evaluating the sustainability and maintainability of projects and dependencies. Depending on a customer's concerns, some metrics are more relevant than others, but there are three categories of metrics.

1. **Community sustainability indicators,** such as the growth of newcomers.
2. **Process-oriented metrics and good practices,** which includes average lead time and review efficiency.
3. **Maintenance metrics,** such as the backlog management index (BMI).

Knowing how healthy (or not!) your projects and dependencies are helps you to make critical decisions about where to invest your attention and resources. It also helps you get ahead of costly problems before they arise.

Think of it like this: you wouldn't buy a car just because it has a shiny paint job. Rather, you'd pop the hood, check the engine, and make sure it's built to last. If a car's essential parts are unhealthy or if there are no shops where you can maintain it, you would look for another car. It should be the same with software development projects.

To carry the metaphor forward, vulnerability scans (the security pillar of open source strategy) are like the car dashboard warning lights. They alert you to issues that already exist. This is important, but it's certainly not everything. When you prioritize project health, however, it's like doing upkeep on the car over time. If you hear a squeak as you go over a speed bump, for example, it's better to get ahead of the problem before parts begin to fail and you end up spending more to fix it. This kind of upkeep helps you to **prevent problems before they arise**. And it helps you know when it's time to trade in an old, under-maintained wreck for something new.

A Project Health Check Reveals Actionable Insights

Having data on project health is empowering. **Knowing where an OSS dependency is weak allows you to take targeted action.**

With project health data and analysis, organizations may choose to invest in the health of a project by dedicating employees to help a project with sustainability, maintenance, or resilience. Organizations may choose to provide directed funding to support remediating any identified weak areas. Or they may choose to abandon and replace a risky dependency with a healthier one.

Organizations may also choose to accept the risk and not do anything. This, however, has been the default option for too long because organizations were not paying attention to project health. That's why legislators are now regulating software production. And still, too often organizations are not making open source decisions based on data. They're walking with their eyes closed and hoping it's in the right direction. They cannot trust their dependencies, because they don't know the health of their dependencies. That car that is not being maintained may suddenly break down.

That's why we love data. Data and insights about the sustainability, maintainability, and overall resilience of projects allows you to make decisions with your eyes open, decisions that let you move forward with trust in the dependencies you've chosen.

Bitergia Radar Is a Solution to Trusting Your Dependencies

What Is Bitergia Radar?

We designed the Bitergia Radar Services to provide an in-depth understanding of project health risk factors to drive decision-making. With software threats on the rise and with an increased focus on software sustainability, we saw how knowing their project health risk can protect organizations and strengthen their software supply chains.

Just like a real radar makes visible what is otherwise hidden, **our Radar Services give visibility into the opaque world of open source software development.** We use data and metrics to offer an easy-to-understand analysis of risk from a project health perspective. Further, the expert analysis and insights we provide help organizations know where to invest their attention, time, and resources.

Miguel Ángel Fernández, one of the data scientists developing Bitergia Radar, explains the approach of detecting project health risk: "The idea is to take the concept of 'code smells'–indicators that point to buggy parts of code–and apply it on a community activity level." So "community smells" are metrics that point to parts of the community that may not be working well or that could be improved. "By assessing these 'community smells,' Miguel Ángel Fernández concludes, "we can identify projects that could have issues in the future." In other words, we identify the riskiness or sustainability of projects based on project health indicators.

Our two Radar Services approach project health from different angles: **While *Bitergia Radar Reports* delve deep into metrics and analysis of a single open source project and its ecosystem, *Bitergia Risk Radar* zooms out to analyze an entire software supply chain at scale, directing attention to the riskier dependencies.**

In the next section, we consider how this approach helps organizations get ready for CRA compliance. Then we detail each of these services and what they can offer to organizations looking for open source risk assessment solutions.

CRA Readiness for Any Organization

A major reason to embrace project health and get insights on OSS dependencies is to comply with the CRA. Compliance is no small task. For example, organizations are now responsible for providing long-term support and updates. That's why the project health analysis that Bitergia Radar provides is so important: current indicators of active and effective maintenance tell the story of riskiness or sustainability into the future. When you know that a critical dependency is becoming unmaintained, for example, you can take action before a bigger problem arises.

CRA readiness includes aligning practices early, even as the specific standards are still under development. At the time of writing, the CRA was enacted with a 3-year timeframe for everyone to work towards compliance. But there are still unknowns. One big unknown, for example, is the outcome of the 41 items in the CRA that the European Committee sent to European standards bodies. The to-be-developed standards should help simplify conformance. Even as we wait for these new standards–and, thus, more certainty on what compliance with the CRA actually means–companies can focus on the general guidelines that the CRA has set forth. But many organizations are at a loss as to how to actually do this.

Bitergia Radar provides a solution to one of the key guidelines of the CRA: the requirement of "due diligence of manufacturers that integrate free and open source software components" (section 21).

Bitergia Radar can help with proactive steps to get ahead of this legislation, including:

- Monitoring the decline of communities
- Ensuring efficient maintenance of projects
- Detecting early warning signs

Software supply chains are complex, but Bitergia Radar was designed as a simple solution to get visibility and insights into project health risk to be able to make proactive, data-based decisions that align with CRA and other regulatory requirements.

Bitergia Radar Reports: A Deep Dive Into Your Open Source Dependencies

Expert Data and Analysis Help Drive Decisions

Bitergia Radar Reports offer the data, analysis, and insights organizations need to drive decisions **to invest, adopt, or abandon a specific open source project or ecosystem.**

Here is a breakdown of the different types of analysis the Radar Reports provide:

Assessment of Community Activity: A developer survey shows that 80% of developers see "maintainer responsiveness" and 86% see maintainer "activity volume" as important when choosing an open source project. Bitergia Radar assesses activity levels, responsiveness to feedback, and the effectiveness of code reviews to see the pulse of the community and to ensure the project thrives.

Analysis of Contributor Dynamics and Collaboration: The more people and organizations contributing to a project, the stronger the project is moving forward. We analyze contributor dynamics and identify opportunities to enhance collaboration and knowledge sharing within the project. This knowledge helps organizations to evaluate sustainability and to optimize their team's contribution strategy.

Detection of Silos: Silos are areas of the project where contributors are not interacting with one another. They are a potential source of risk because it means there is a lack of coordination and alignment, an early warning sign of a declining project.

Evaluation of Component Health: We evaluate key components, and analyze the level of project health risk based on important metrics. You can understand not only the overall health or riskiness of a project, but also see how each part is doing. This understanding can help you to make informed decisions about resource allocation.

Competitive Analysis: We contrast the health of the focal dependency with that of others that provide similar functionality. Contrasting project health with competing alternatives can reveal the relative competitiveness of an open source project. A project with warning signs may still be the best option compared to its alternatives.

Corporate Backing: Much open source is developed by and for companies. To understand the health of an open source project, one must know who the corporations are that influence its development and roadmap. A single vendor project, for example, has a higher risk of relicensing.

Too often, critical choices regarding open source software are made without a thorough understanding of the underlying risk factors. We believe that robust data and insightful analysis are key to successful decision-making.

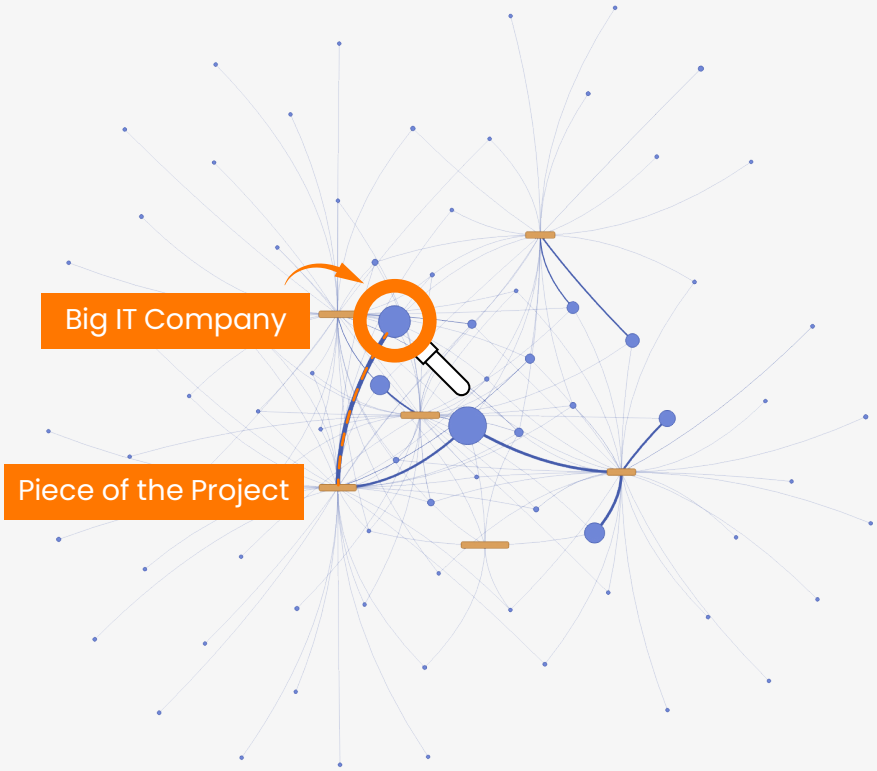
Explore Radar Reports

Case Study: Resolving Doubts about Projects into the Future

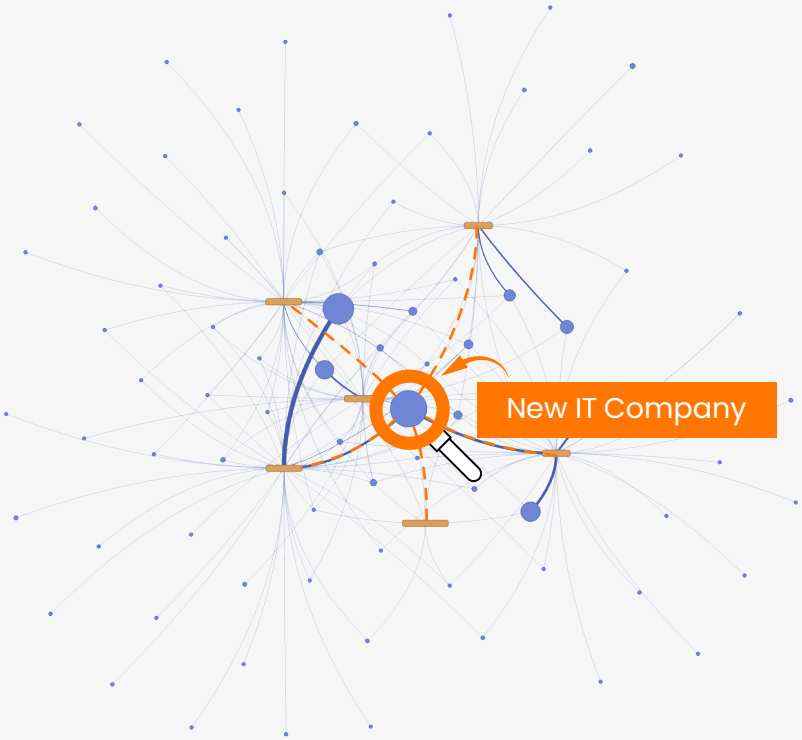
One Bitergia Radar Reports customer was concerned about the riskiness of a dependency they relied on. They were able to use the data and analysis we provided to alleviate their worries, make a confident decision about the project, and save money.

This customer was concerned: they'd heard that a major IT company was leaving the project, and they wanted to know their risk.

The Radar Reports analysis revealed that this company was indeed leaving the project, but that their contributions had been small to begin with, only working towards maintaining one repository. Their leaving would have little impact.



On the other hand, our customer was happily surprised to learn that another large company had joined the project, and was leading the maintenance across repositories. They were pushing the project forward, and helping ensure its sustainable into the future.



With this information, our customer could move forward confidently. They decided to reach out to this new company to establish a collaboration. They also had the information they needed to trust in the project they were using. There was no need to make a costly switch to something different.

Trust the Projects You Rely On

Too many organizations make important decisions without data to back them up. Bitergia Radar Reports offer a solution to trusting specific open source dependencies.

When software resilience is more important than ever, this service provides project health metrics and insights to confidently invest in, adopt, or abandon open source projects, minimizing risk and maximizing profitability.

VI Bitergia Radar

Bitergia Risk Radar: An Analysis of Risk in Your Software Supply Chain at Scale

This Risk Assessment Solution Is Unique

While the Radar Reports focus on one single project or ecosystem, Bitergia Risk Radar provides **a sweeping analysis of your software supply chain at scale.** We evaluate the sustainability of dependencies by using SBOMs as inputs.

Bitergia Risk Radar is unique. We identify problematic software development practices before problems manifest in the software. While other models analyze the software code itself (software composition analysis is important!) or good development practices, we are looking at the problem from a sociotechnical or developer community perspective, asking questions like:

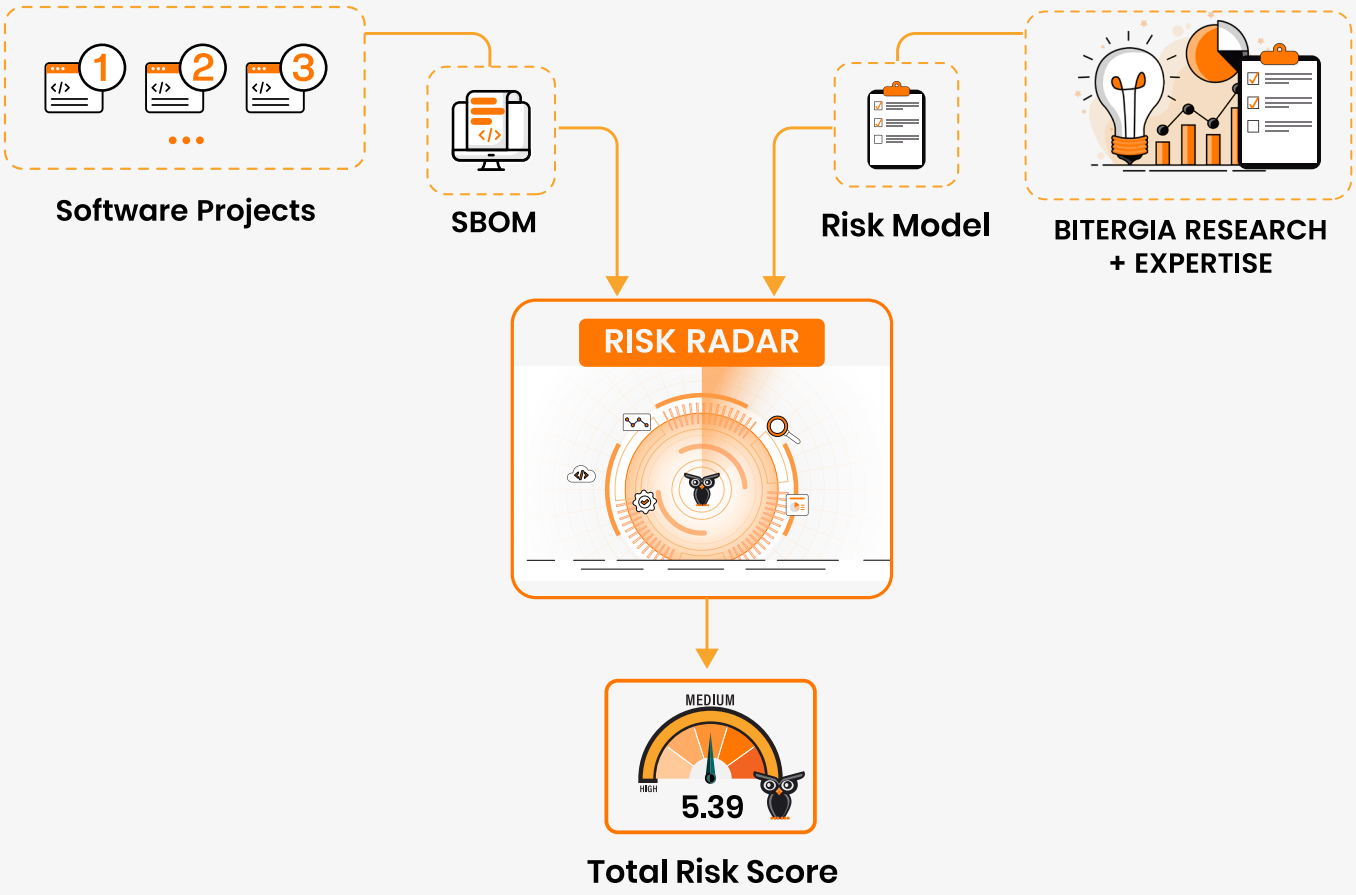
- Is the project actively maintained with a dedicated community?
- Does it attract new contributors who will carry it forward?
- How efficient and effective is the open source community in addressing its issues and change requests?
- Is it viable for the long term?

At the very core of Bitergia Risk Radar is the idea that a well-maintained open source project is less risky and more sustainable. A project that is not actively or effectively maintained is unlikely to fix vulnerabilities in a timely manner. A healthy project, on the other hand, establishes best practices to prevent vulnerabilities from becoming manifest, for example, by practicing rigorous code reviews. By identifying and addressing “community smells” early, you can have more trust that you will not be blindsided by an attack and that your software will last over the long term.

We have two practical aims in creating Bitergia Risk Radar: manage and reduce risk from using OSS dependencies, and provide user-friendly results. We want to offer a solution that gives organizations predictability, control, and trust in their dependency use.

We understand the complexities involved in understanding dependency use. That's why we aim to make the process as easy as possible.

What Does the Bitergia Risk Radar Process Look Like?



- As the visual shows, we first work together to identify all of the OSS dependencies used by your organization's software components.

- If you do not yet generate a Software Bill of Materials (SBOM), you will want to do so anyway to be ready for complying with the new European Cyber Resilience Act (CRA), and we can help establish the detection of OSS dependencies in use.

- We then run the SBOM and the dependencies' data (their repositories and meta-data) through the Bitergia Analytics Platform. Specifically, we feed the risk model with data from data sources such as Git, GitHub, and GitLab. The Bitergia Analytics Platform provides us with analysis and metrics on the developer communities that maintain these dependencies.

- Armed with the resulting metrics, we pass them through the risk model. In this way, **all the metrics are calculated into actionable knowledge about risk for each dependency.**

- The result of this process is a simple **Total Risk Score** for each dependency. The score is easy to read, interpret, and communicate. It directs your attention to areas of risk and helps you to take targeted action.

- You will have access to the **Bitergia Risk Radar Dashboard**, where you can access the risk scores and visualizations. Importantly, you can also drill down into the individual metrics to pinpoint where the risk really lies.

- For developers, the Total Risk Scores can also be integrated into the CI/CD pipeline to be presented in the development workflow.

- We recognize that each company has unique needs. Our consultancy team works with you to customize the scoring weights, select the most impactful visualizations, and establish a repeatable reporting format. We meet stakeholders where they are and support their decision-making with quality data.

These 7 Metrics Reveal Early Indicators of Risk

To give a concrete example of a risk model, we can predict the risk level of an OSS dependency with **7 risk metrics**. Each metric analyzes the social developer activity in maintaining the OSS dependency from a distinct perspective. The 7 metrics can be divided into 3 categories of maintenance issues to watch out for. Here are the 3 categories and the metrics they encompass:

1. The community cannot handle demand:

- Backlog Management Index (BMI)
- Review Efficiency Index (REI)

2. The community does not address work quickly:

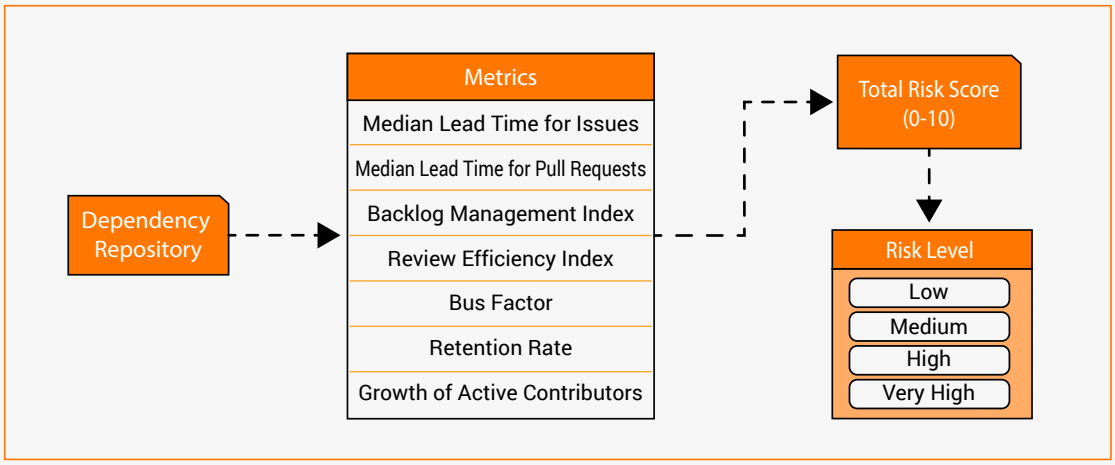
- Median Lead Time for Issues
- Median Lead Time for Pull Requests

3. The community lacks sufficient attention:

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

Each risk metric has specified thresholds to determine the Risk Level. As the image depicts, the individual risk metrics are combined into a Total Risk Score from 0-10–or from low to very high risk. The score is easy to read, interpret, and communicate. A very high risk score, for example, is the result of a dependency having very little maintainer activity or none at all. This is an important "community smell," because zero maintenance is inherently risky. It leaves a dependency open to future vulnerabilities.

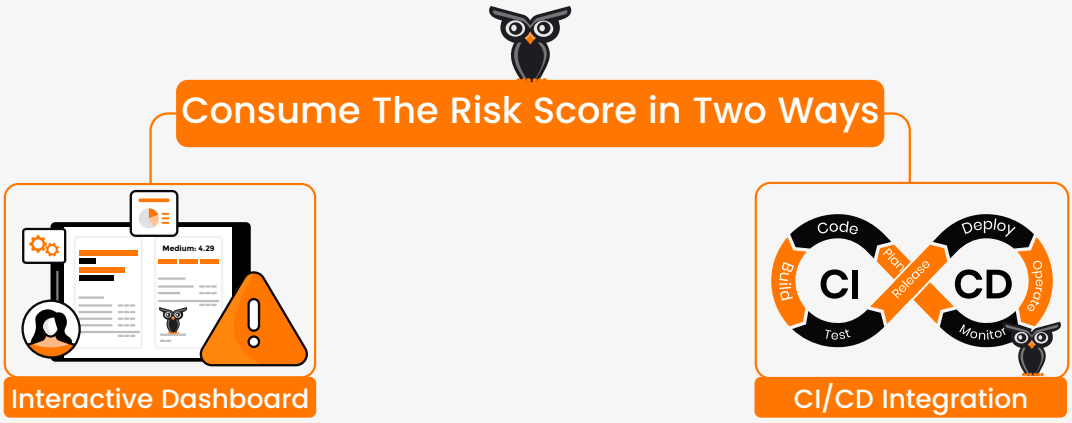
The Total Risk Score gives users a good overview fast, and helps direct attention to where it needs to be. Then users can drill down into the 7 risk metrics to analyze how each social aspect of the developer community informs the Total Risk Score.



Consume The Risk Scores in Two Ways

Bitergia Risk Radar offers two ways to consume the risk analysis. From a management and risk controller perspective, we offer a dashboard. The Risk Radar Dashboard allows users to see the aggregate risk evaluation for the entire set of dependencies. They can then filter down to focus on those dependencies with the highest Total Risk Score.

The second way to consume the risk score is for developers. Through an API, the risk scores can be presented during the development workflow as part of the CI/CD pipeline. Risk is most effectively managed by surfacing it early (by "shifting left"). The earlier risk is detected, the better able organizations are to get ahead of costly attacks and to ensure the sustainability of their software.



Case Study: Filling a Gap in the Current Risk Assessment Landscape

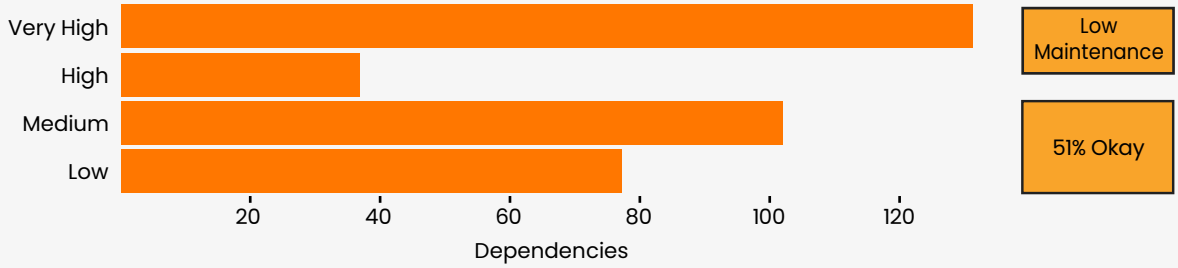
The development of Bitergia Risk Radar began with a need in the open source security space. Specifically, it began when one of our customers realized that the software supply chains that their organization and others relied on may not be as secure as they had thought. For them, the Log4j vulnerability was a wake-up call.

We worked with this customer to develop Bitergia Risk Radar and to fill the gap in the current risk assessment landscape. Bitergia Risk Radar goes beyond traditional SBOM and SCA approaches that evaluate the software licenses and source code. Instead, it builds trust by looking at project health and maintenance activity for early indicators of risk.

To demonstrate this approach, we ran the model against a subset of the Kubernetes dependencies. More specifically, against those dependencies' last 12 months of activity (from June 2023 to June 2024). As Kubernetes is a large and highly trusted development project, Bitergia was able to use it to fine-tune the model and to get a large sample of results.

These results were illuminating. 179 of the 347 dependencies analyzed (about 51%) got a Total Risk Score of low to medium risk. However, 37 dependencies got a high risk score, and 131 showed as very high risk. A very high risk score is the result of a dependency having very little community activity or none at all. Miguel Ángel Fernández, data scientist at Bitergia, was surprised by this result: "I expected some [unmaintained dependencies] in such a large project as Kubernetes...but not so many."

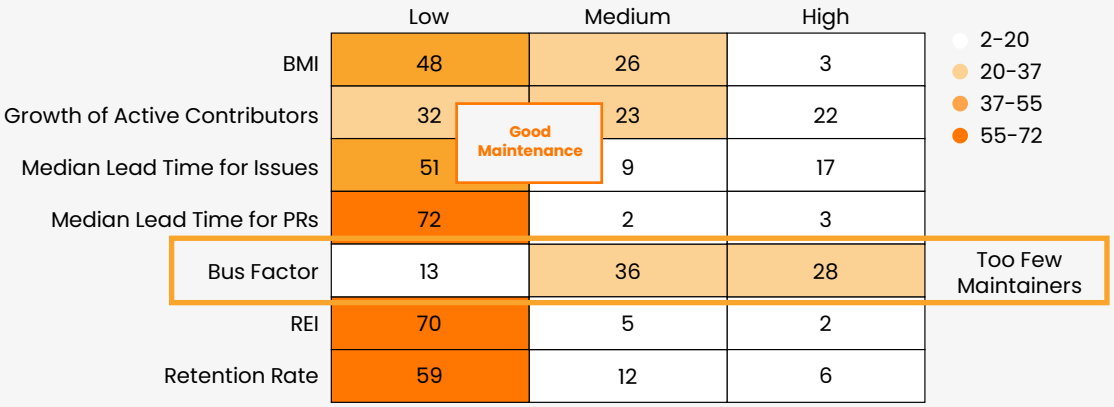
Dependencies by Risk Value



Further, even in the dependencies with a low Total Risk Score, Miguel Ángel Fernández found that the individual Metrics with a higher risk were related to the number of people maintaining the code. "The Bus Factor," for example, is a metric that identifies the number of people contributing 50% of a project's code. According to this metric, five or more contributors is low risk, two to five is medium risk, and only one or two is high risk. Ideally, a project has more than five people contributing half of the code. But this is usually not the case.

28 of the dependencies with a low Total Risk Score still had a high risk in the Bus Factor. If these one or two maintainers leave the project, and others do not replace them, they take all of the knowledge of the project with them. It may then be left open to future vulnerabilities.

Risk Value per Metric, by Number of Dependencies



Individual Metric Risk Value

These are some of the most interesting results from this experiment with Kubernetes, although this process yielded a great deal more information that Bitergia has used to fine-tune and focus their model and analysis.

As the complexity of software supply chains continues to grow, the need for effective security and risk management solutions will only become more critical. Bitergia's work with Kubernetes demonstrates the power of a project health approach in addressing this challenge.

Bitergia Risk Radar fills the gap in software supply chain security, assessing risk factors early and at scale, before those factors turn into costly attacks.

Take Control of Project Health Risk

The resilience of any software project depends on the health of all of the dependencies holding it up. In the modern software development landscape–filled with threats and new regulations–being able to trust those dependencies is more important than ever. It is also a major challenge. That is why we developed Bitergia Radar.

We saw that data, analysis and insights about project health can alert organizations to early indicators of risk so they can make decisions that fortify the sustainability of their projects, and so they can trust that their supply chain is strong. With Bitergia Risk Radar and Bitergia Radar Reports, organizations can now proactively address potential threats and make informed and profitable decisions about risk in their open source dependencies.

Bitergia

Let's Build Trustworthy Projects Together. Reach out to Learn Your Project Health Risk!

- info@bitergia.com
- LinkedIn
- Read more about Bitergia Radar