Building a Software Chain of Custody

A Guide for CTOs, CIOs, and Enterprise DevOps Teams

XebiaLabs
Enterprise DevOps

White Paper

Building a Software Chain of Custody

A Guide for CTOs, CIOs, and Enterprise DevOps Teams

Executive Summary

In today's software-driven world, organizational leaders cannot ignore the fact that hackers, viruses, malware, data breaches, and compliance violations all threaten the integrity of their software assets. Compromised software integrity can lead to operational shutdowns, contract breaches, lawsuits, and huge fines that affect revenue and profitability, damage corporate credibility, and, ultimately, cause irreparable harm to brand reputation. It's nearly impossible to quantify the long-term cost of the consumer and investor trust that is lost after an organization is in the news for a software hack or data breach.

A Software Chain of Custody is key for proving the integrity of software assets throughout the enterprise. In the legal world, the chain of custody for a piece of evidence ensures the integrity of that evidence; similarly, the Software Chain of Custody proves what happened, when it happened, where it happened, and who made it happen during the software delivery process—from the time you plan features through the time they're deployed in production. A Software Chain of Custody also feeds valuable contextual data into Value Stream Management, so leaders can analyze and continuously improve software delivery processes.

This white paper explains how you can build and benefit from a Software Chain of Custody that automatically captures and contextualizes the critical data you need to prove the integrity of your software planning, development, and delivery processes, from end to end.

Asset Integrity in a Software-Driven World

Today, it's widely accepted that every business is a software business, which means that every large enterprise that builds, buys, or runs software—from leading retailers, to financial service providers, to insurance companies, and more—must be concerned with the integrity of its software assets. Software asset integrity is everyone's responsibility: from C-level executives to product owners, release managers, auditors, and DevOps team members. All stakeholders need visibility into planning, design, build, test, release, and monitoring processes, so they can prove that the software running in their environments is truly what it claims to be: it's the right version, it contains the code it should, it uses secure and approved libraries, and it hasn't been compromised by a virus or malware.

Software asset integrity is heavily influenced by aspects of software quality such as:

- **Traceability—**Is every software artifact stamped with a unique identifier that can be verified as the artifact moves through environments?
- **Performance—**Does the software perform as it should? Can you detect whether performance is degrading over time?
- **Horizontal scalability—**Can you increase capacity on the fly by adding physical servers, virtual machines, container instances, or pods?
- **Security—**Is the software protected from data breaches and other security violations? Can you detect if and when security is compromised?

Software asset integrity is inexorably linked to the credibility of your corporate brand because the consequences of operating low-integrity software are severe. Compromised applications can lead to lost business income, operational shutdown, and breach of contract. Data storage that is not properly secured is vulnerable to breaches that expose sensitive information. Some government regulations, such as the United States' Sarbanes-Oxley Act, can even carry the penalty of prison time for organizational leaders.

Not Securing Your Software Can Be Expensive

Consumers worldwide share a growing concern about how their personal data is secured. Organizations must prioritize compliance with government regulations designed to protect individuals and prevent fraud. Recent high-profile fines for compliance violations include:

- $575 million for exposing 147 million people's personal information
- $230 million for exposing 500,000 customers' credit card data
- $148 million for failing to disclose a breach of 57 million user accounts

Organizational leaders and DevOps teams struggle to identify and mitigate risks that endanger the integrity of their software assets, because the work required to do so is time-consuming, largely manual, and prone to errors. Executives, VPs, and directors don't have the visibility to be sure that security and compliance needs are being met. Product owners and DevOps team leads must figure out how to satisfy audit and compliance requirements that threaten to slow down their teams. Software engineers spend time collecting raw data from technical tools and delivering it for audit reports instead of building value-adding features. And security, audit, and compliance groups are stuck going back and forth between teams, trying to get the information they need.

Enterprises Need a Software Chain of Custody

The concept of a chain of custody originated in the legal world, where it describes the way a piece of evidence for a legal case is handled, transferred, stored, and analyzed. Today, many industries have adopted the concept to describe the way that processes, and the people who are involved in them, should be tracked and documented. For example, in logistics, the chain of custody represents the path that a product takes from the start of the supply chain to the point where the consumer can buy it. In food production, the chain of custody represents the path that a food product takes from raw ingredients to the final package on the grocery store shelf.

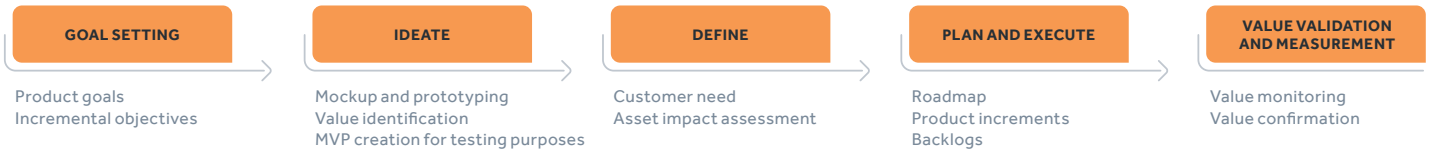
The chain of custody concept can also be applied to software assets in your organization. Just as a chain of custody ensures the integrity of a product or a piece of evidence, the Software Chain of Custody proves what happened, when it happened, where it happened, and who made it happen during the software delivery process—from the time you set business goals and plan features, all the way through development, testing, deployment, and monitoring of live software in production.

> A Software Chain of Custody that tracks and documents every step of the software delivery process proves the integrity of your software assets.

Building Your Software Chain of Custody

A Software Chain of Custody that truly covers the enterprise-scale software delivery process starts with the strategic goals that you want to achieve. Most organizations execute a pipeline of business activities similar to the one below when determining which software assets to invest in and how to measure their return on investment.

A typical pipeline of business activities



The inputs into and outputs of this process are usually scattered in many places. Organizational leaders have ideas in mind and sketch out vision statements and goals on whiteboards; product managers build roadmaps in PowerPoint; release managers create plans and schedules in Excel; and product owners add work items to a backlog in Jira. A Software Chain of Custody can transform this disconnected set of inputs and outputs into a structured, connected, traceable chain of decisions, activities, and outcomes.

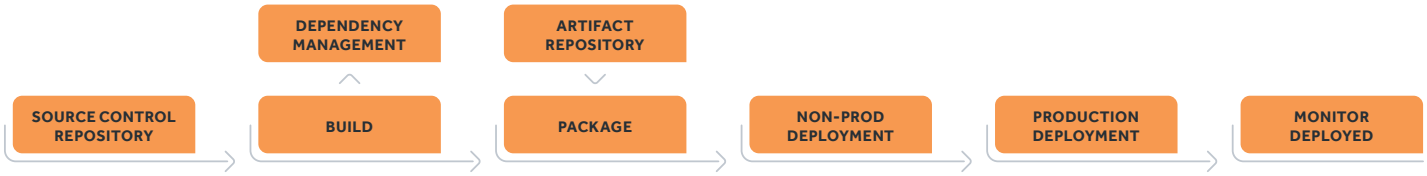
Portfolio Planning and Agile Management tools such as CollabNet VersionOne are rich sources of data for the activities in this business-oriented process. Capturing that data and building it into your Software Chain of Custody gives you a complete picture of your true end-to-end process so you can:

- **Track compliance tasks that happen outside the technical pipeline**
- **Better visualize and understand your value stream**
- **Use data to align the backlog of technical work with strategic goals**
- **Measure DevOps teams' performance against strategic goals**

Understanding the Challenging Landscape of DevOps Tools

IT Revolution's *DevOps Automated Governance Reference Architecture* (September 17, 2019) illustrates the technical software delivery pipeline as follows.

The technical software delivery pipeline



DevOps teams manage and execute the activities that move software assets through the development and delivery process: writing new code, integrating it with the codebase, testing it, packaging it into applications, deploying those applications to both pre-production and production environments, and monitoring each application's availability, stability, and performance. These activities require a variety of tools that support source code management, continuous integration, build capabilities, environment provisioning, application deployment, log aggregation, and performance monitoring.

This landscape of varied, often disconnected tools makes it inherently challenging to create a Software Chain of Custody that reliably tracks the work that is being done and that captures and documents which person or process triggered the work.

> DevOps teams are often required to manually collect, compile, and correlate data across tools to satisfy audit reporting requirements—a process that is time-consuming, prone to human error, and that distracts developers from the work of building value-adding business applications.

Automation is Key for a Scalable, Repeatable Software Chain of Custody

To establish a Software Chain of Custody for technical software delivery processes, you must automate the process of capturing and correlating evidence for delivery activities across all tools in the DevOps landscape. According to *DevOps Automated Governance Reference Architecture*:

> As more and more DevOps practices are automated, it becomes harder to capture the data required to ensure all security and compliance concerns are met. Organizations need an automated way to track governance throughout the entire software delivery process so they can attest to the integrity of all assets and to the security of all running applications.

Implementing this type of automation, and then optimizing it for enterprise-scale use, requires tools and mechanisms that are built for repetition and scale. A Software Chain of Custody process that is highly repeatable ensures that audit evidence is captured for every release. This process must scale to capture evidence automatically for every change made to every software asset—no matter how many tools are involved or how complex the technical delivery pipeline is.

Continuous Improvement through Value Stream Management

Value Stream Management is an emerging discipline that tracks software delivery activities and provides the contextual data enterprises need to analyze and continuously improve their software delivery processes. A Software Chain of Custody provides the foundation for Value Stream Management because, in addition to proving the integrity of software assets, it paints a complete picture of each and every software release from beginning to end.

In many enterprises, DevOps leaders use strategic goals to build a roadmap that guides decisions about the future of software assets: which assets to build, which assets to change, and even which assets to retire. Product managers break the roadmap down into portfolio items, grouped by strategic theme, that turn goals into actionable work. Then, product owners create a backlog of user stories and changes so teams can plan and estimate their day-to-day workload and delivery timeline.

Many enterprises work this way, but they lack a rich, end-to-end chain that would help leaders understand whether the technical changes that are being implemented actually support the company's strategic goals—especially when the software delivery process happens in long-running, complicated delivery pipelines that span many applications or teams. Value Stream Management that's built on top of a Software Chain of Custody provides the crucial data and insights needed to identify opportunities for automation, reduce release delays, eliminate process bottlenecks, and, ultimately, help teams push changes to production faster.

The value stream tracks software delivery activities for continuous improvement

How to Take Control of Your Software Assets

Capturing audit evidence in a scalable, repeatable Software Chain of Custody is only one piece of the governance puzzle for technical software delivery processes. There are other impactful changes you can make to ensure that the lifecycle of your software assets is properly controlled, governed, and documented—without burdening DevOps teams with cumbersome checklists, sign offs, and other manual work that slows down software delivery processes.

Simplify the Compliance Framework

Many organizations require DevOps teams to provide input for audit reports and perform other compliance-related tasks that are based on outdated or poorly understood requirements, or that can be better satisfied by applying modern automation techniques. Simplifying and streamlining the compliance framework helps DevOps teams better understand what is required of them, so they can more quickly implement automated controls that ensure software delivery pipelines are compliant by default.

Shift Validations Left and Automate Them

Shifting security and compliance validations left means executing them as early as possible in the software development process—typically by implementing them in the automated test phase. There are many open source and commercial tools available that support automated security and compliance testing, such as Chef InSpec, SonarQube, Black Duck, Checkmarx SAST, and Fortify Static Code Analyzer.

When you shift security and compliance validations left:

- DevOps teams can identify security and compliance problems earlier in the software delivery process, when there is time and capacity to fix them
- Software developers become more cognizant of security and compliance requirements and the consequences of not meeting them
- The results of automated security and compliance tests become an immutable part of the Software Chain of Custody
- The feedback loop between development, operations, security, and audit teams becomes stronger

Build Continuous Verification into the Process

Continuous verification of the activities in the software delivery process helps ensure that all software assets go through the tests and checks that are required to ensure their quality and security. In *Improving DevOps through Continuous Verification* (January 21, 2020), David Jasso of VMware explains how you can build continuous verification into the technical software delivery pipeline by adding governance checks that perform functions such as:

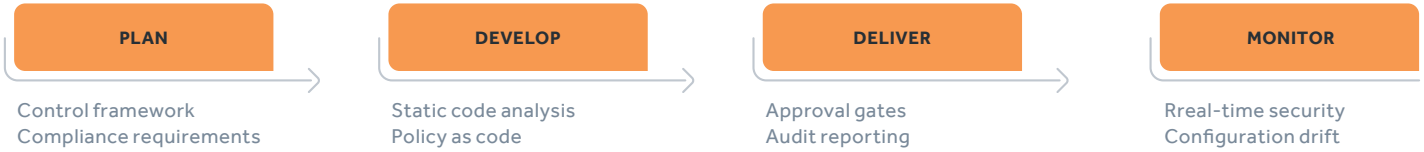
- Verifying the expected resource utilization and associated costs of those resources won't exceed authorized limits
- Validating that the configurations of the IaaS and PaaS resources they're using meet specific security and compliance best practices
- Ensuring that once deployed, an application can achieve specific performance thresholds

The results of these governance checks help ensure the integrity of software assets as they move through he development, build, test, and deployment process—even when that process includes many different tools and people.

Collect and Store Governance Data Throughout the Process

In addition to the technical software delivery pipeline, there is a pipeline of governance activities that are required to prove that the work that is done in the technical pipeline satisfies industry and regulatory requirements.

The pipeline of required governance activities



It is crucial that governance data is automatically collected and stored throughout the software delivery process, because that data is the input required for reports and metrics that support the Software Chain of Custody.

Governance data is also a useful resource for DevOps teams that want to continuously improve their technical pipelines by automating time-consuming manual tasks, identifying and reducing rework, and locating and eliminating bottlenecks.

XebiaLabs Powers the Total Software Chain of Custody

The XebiaLabs DevOps Platform powers the total Software Chain of Custody by integrating with tools throughout the DevOps tool landscape, orchestrating the technical software delivery process, and automatically collecting relevant data from the beginning to the end of the process. XebiaLabs' industry-leading Release Orchestration capabilities allow DevOps teams and engineering leaders to make continuous security and compliance testing a native part of the software delivery process.

As an end-to-end DevOps Toolchain Orchestration and Reporting platform, XebiaLabs is in the unique position to capture data across all tools, provide context for release activities, and tell the story of exactly what happene in each release. The XebiaLabs DevOps Platform doesn't just trigger tests and verification checks in other tools; it also harvests the most relevant data from those tools and automatically builds an easy-to-understand picture of each and every software release. XebiaLabs combines portfolio and backlog planning information from Agile Management tools such as CollabNet VersionOne with technical data from build, test, deployment, and monitoring tools to create a context-rich Software Chain of Custody that benefits technical and business stakeholders alike.

With the XebiaLabs DevOps Platform:

- Development leaders can build guardrails that ensure compliance and security tasks always happen as part of automated release processes
- DevOps teams get instant, automated audit reporting so they can prove compliance, predict and mitigate release risk, and release better software, faster

- Product owners, release managers, and engineers no longer waste hundreds or thousands of hours piecing together data and creating incomplete reports
- InfoSec teams, auditors, and other business stakeholders get the information they need, in the format they need it, instantly
- CIOs and CTOs can rest easy knowing that governance and security processes have been followed and compliance is easy to prove

Enterprises Need Push-Button, On-Demand Audit Reporting

XebiaLabs' on-demand release audit report powers the Software Chain of Custody by covering all activities in the enterprise software delivery pipeline. The audit report provides evidence for every manual and automated task in the software delivery process at the push of a button: who did what, when, where, and how. Create report filters by date, folder, keywords, and more, and export information for one or many releases. Drill down on release tasks and jump directly to the tool of origin for details.

Summary

To secure software assets, comply with governance requirements, preserve profitability, and protect brand reputation, organizational leaders in every industry need a Software Chain of Custody that covers the end-to-end software delivery process: from ideation and planning, to development and testing, to deployment and monitoring in production. A Software Chain of Custody gathers the evidence needed to prove the integrity of software assets and provides a foundation for Value Stream Management that enables continuous improvement of software delivery processes. The XebiaLabs DevOps Platform automatically delivers a comprehensive Software Chain of Custody that provides the platform you need for fast, secure, compliant software delivery. And you'll be able to prove it. Visit xebialabs.com/compliance and xebialabs.com/software-chain-of-custody to learn more about taking control of your release processes and harnessing the power of automated audit reporting.

About XebiaLabs

The XebiaLabs DevOps Platform helps businesses achieve Continuous Delivery by automating and accelerating complex software delivery pipelines, while enabling DevOps teams to keep using the tools they love. With XebiaLabs, you can connect your DevOps tools, orchestrate release activities, ensure compliance with IT governance requirements, and deploy applications to any target technology, all without ad-hoc scripting. The platform automatically captures critical release data and lets you generate comprehensive audit reports at the push of a button. Only XebiaLabs unifies DevOps pipeline data to provide a complete system of record for the enterprise software delivery lifecycle.