

Why Release Orchestration Is the Gateway to Modern Software Delivery

Table of Contents

Introduction	3
Making Things Easier for the Release Team	4
The Missing Link in Mature DevOps Organizations	5
8 Benefits to Expect from Release Orchestration	8
Building a Better Value Stream	12
Example Customer Outcomes from Effective Orchestration	13
CloudBees Unify: Streamlined Orchestration for Scalable Software Delivery	15

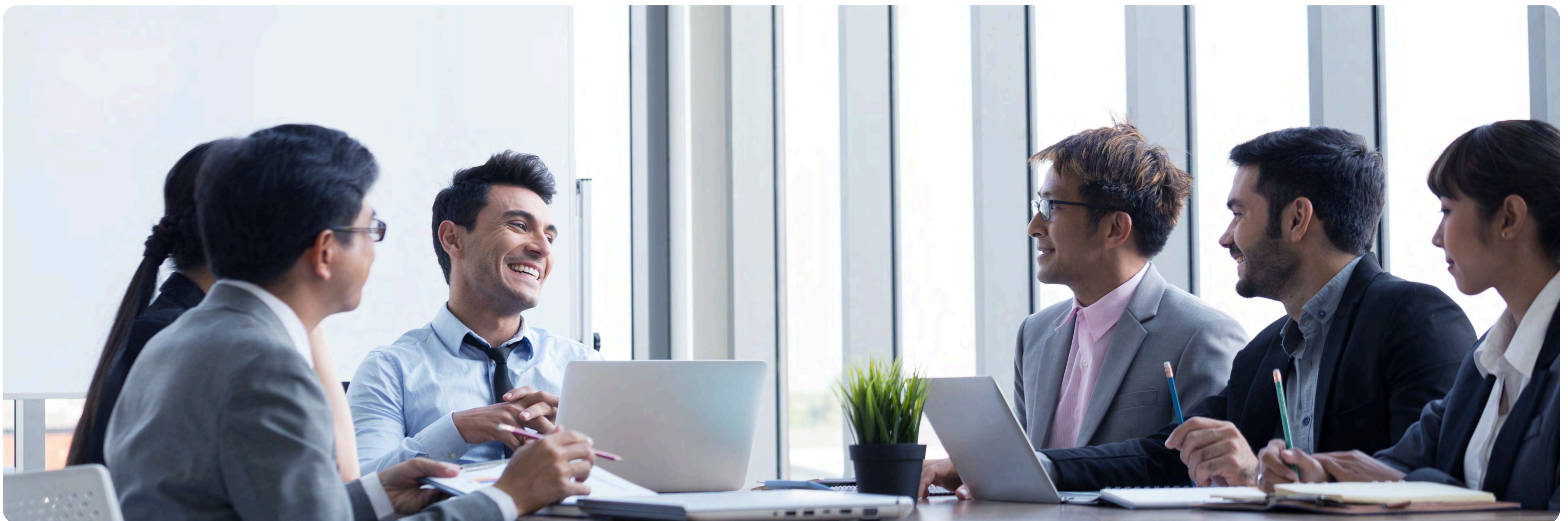
Introduction

The power of automated software development is enticing. The quick wins that organizations get from automating tasks such as infrastructure deployments and software tests and builds are impressive, but they also often prevent teams from seeing further possibilities.

Release orchestration is the next step in software delivery maturity. It provides higher-order visibility for management to see the bigger picture across tools, pipelines, features, releases, and dependencies. It also facilitates team management across the above—regardless of the application or environment type. It connects islands of automation through sequencing and allows teams to benefit from visibility into other parts of the organization and enforce compliant release processes.

This powerful discipline is a critical success factor in delivering maximum value to software customers. It provides an end-to-end customer-centric view that fosters a value-driven approach to DevOps.

Making Things Easier for the Release Team



It takes a cohesive set of skills to safely deliver modern software. An effective release management platform must support them all, encompassing both developer practitioners and the broader release team.

As developers face increasing pressure to consider security issues at every step of software design and development, they need all the support they can get to keep delivering quality software at speed. The release management platform must support the tool sets that developers choose to build their software. It should ideally be invisible to the developer while providing valuable cross-pipeline support in the background, maintaining a premium developer experience.

There are multiple teams involved throughout the release process. For example, your heads of DevOps, site reliability engineers (SREs), and platform engineering teams are typically responsible for creating the release pipeline templates, whereas engineering managers are typically involved in planning the actual software releases and deadlines. They all must be ready to articulate an up-to-date picture of software release status to business stakeholders. Complexity and a bracketed view of individual DevOps stages and environments have limited this team's capabilities.

While developers get to choose their own high-quality tools, release management tooling is often overlooked. This leaves those responsible for managing releases struggling with makeshift software not tailored for the task. They end up keeping track of complex release timelines in spreadsheets or homegrown systems while communicating with stakeholders via email.

These tools don't allow for real-time updates, leaving the release team constantly looking in the rearview mirror when it should be looking ahead. It has a disjointed view of a complex development and deployment process, making it difficult to see ahead and plan for speed bumps. A siloed view of the software development and production process leaves the team in a reactive state, constantly worrying about possible problems such as last-minute bugs or production issues. The community even has a term for this: “release anxiety.” An effective release orchestration tool must balance these requirements, supporting the developer experience while also using intuitive tooling to support the release process and eliminate release anxiety. It must ultimately acknowledge and enhance the business outcomes from effective delivery.

The Missing Link in Mature DevOps Organizations

Many companies equate orchestration with automation. In practice, the two are separate, but complementary, activities.

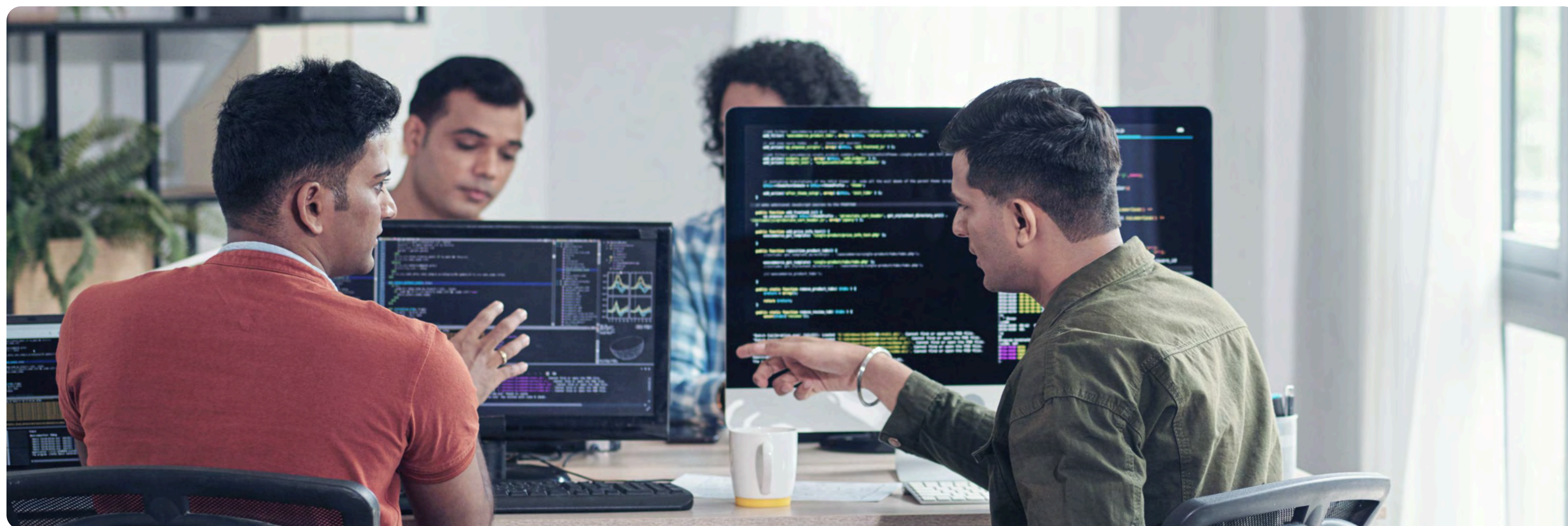
Automation drives efficiency into the delivery process, doing things faster and reducing human error for more reliable results. However, many automated pipelines still rely on manual steps that lie between the automated tasks, leaving teams with a “pay no attention to the man behind the curtain” mentality when it comes to various steps in a release pipeline.

The perceived complexity of orchestrating manual steps alongside automated tasks sometimes prevents companies from automating at all. This is where release orchestration comes in to help make this process easy.

Automation is only part of a broader journey. Too many teams automate tasks without thinking about the part that they play in broader processes. Many companies fail to measure the effect of automation beyond the basic speed and accuracy metrics, which are mere table stakes in any mature DevOps process. Taking the next step is key to delivering deeper value.



In reality, automation and orchestration are separate, but complementary, activities.



Orchestration sequences those tasks into end-to-end process chains in which each automated task communicates with the next. It offers a coordinating layer connecting different tools, pipelines, and processes in a single view. This creates a holistic view of the software development and deployment process across multiple platforms while improving stakeholder communications.

Orchestration also collects data from across those tools and tasks in a single place for analysis. This creates an end-to-end view across the entire pipeline portfolio, allowing team members to see beyond individual pipelines and filling gaps in their intelligence.

8 Benefits to Expect from Release Orchestration

Release orchestration is the missing link for DevOps transformation. Without it, teams will suffer internal friction and see limited results from software pipeline automation. Automation's early benefits will plateau without taking this vital next step. Here are some of the critical benefits that release orchestration delivers:

1 Reduced internal friction

Release managers, developers, and business stakeholders all have their own agendas. Without a unifying platform to coordinate releases, each tends to satisfy their own needs without accommodating others. By providing a higher order of visibility and control over the entire development and deployment process, each of them can meet their own requirements.

Developers can continue coding without repeated queries from the release manager. The C-suite gets custom, real-time reports that show them the insights they need without prompting. Compliance and security teams get the security audits they need before release time. And, the release manager gets to go home on time.

2

Consistency and certainty

Smooth delivery demands a consistent process that is resilient against unexpected problems, such as last-minute bugs. Release orchestration offers this by allowing practitioners to model releases, along with the environments they run on and the processes that support them. These release models can be preserved for reuse, ensuring that all contributors to a release follow the same documented and tested processes.

These models turn pipelines into products in their own right, offering self-service catalogs of automated process chains that provide clear guard rails for development teams to engage in consistent, compliant delivery.

3

Confidence in release decisions

These repeatable pipelines deliver the data that practitioners need to make the final decision when pushing code to production. Because an release orchestration system gathers data from tools across all pipelines and releases, it can document all activities made along the way, ensuring that all the necessary automated and manual gate conditions have been satisfied. This increases release confidence and minimizes the risk of production problems.

4

Confidence in release decisions

Automated tasks that are not connected lead to gaps in release advancement. The hand-off between building and testing software is often manual, as is the running of different tests on dynamically generated environments. Even the approval of those tests and subsequent progression to release involves manual intervention. That leaves release managers unable to know when something will happen, who will be around to handle it, and what upcoming resource constraints that might create.

Release orchestration prevents that by visualizing data that exposes these hidden risks, including the time spent in each stage and how much of it was automated. It helps practitioners to see and rectify emerging bottlenecks in the end-to-end process.

5

Early problem detection

Practitioners get full visibility over the entire software development process allowing them to spot and handle emerging issues before they become full-scale problems.

6

Security and compliance

Release orchestration allows you to converge on three key components of security:

- **Visibility** - the release pipeline offers a central point of insight into your automated and manual processes.
- **Traceability** - it enables security teams to better track automated and manual tasks in the development pipeline, making it useful for root cause analysis.
- **Compliance** - this visibility and traceability makes it easier to keep the development process compliant. An release orchestration system is aware of all the steps in your release process, and documents them so that security teams and auditors can quickly assess which security decisions were made, when they were made, and by whom.

Fine-grained access control for everyone involved in a release, including automatic or manual authentication, contributes to a secure software development lifecycle. Tools integration leads to a data-driven process that documents every action and its instigator to satisfy audit and governance requirements.

7

Better
intelligence
through
analytics

Release orchestration provides higher-order visibility over the development process via a centralized dashboard, giving practitioners a bird's-eye view of release health across the entire technology portfolio. This enables them to quickly analyze their key metrics, ranging from resource usage to release throughput, which they can relay to key stakeholders at short.

8

Scalability and
flexibility
across cloud
environments

Organizations might get away with discrete automation when working on relatively small technology portfolios, but as they scale up their activities, the gaps between automated tasks will begin to show. An effective release orchestration platform must scale up with cloud environments, supporting Kubernetes-based infrastructures for handling large numbers of containers at scale. It must span application development models from traditional monolithic applications through to microservices across multi-cloud and on-premises environments.

Building a Better Value Stream

Beyond these important tactical gains, release orchestration offers a broader strategic benefit that underlines its importance to the software function and its stakeholders: contribution to the value stream.

The value stream is a series of steps towards providing value to an end customer through code delivery. It connects and measures customer value across the entire software delivery process, from initial feature request to value delivery to the customer. This isn't something a team can realize by focusing on discrete task automation. It takes a unified platform that offers high-level release visibility.

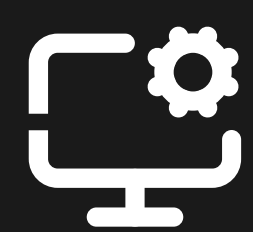
Effective release orchestration enables value stream management (VSM) by providing the intelligence needed to see beyond how an organization automates aspects of the delivery process to concentrate on the “why.” That frees software teams to think about what value they want to deliver to the customer, how to improve the processes that support that delivery, and how to measure those improvements. It creates robust metrics they can use to satisfy customers and enhance that value over time.



Effective value stream management (VSM) means focusing beyond how an organization is automating aspects of the software delivery process to concentrate on the “why.”

Example Customer Outcomes from Effective Orchestration

Release orchestration can help to deliver extra value to business stakeholders and make life easier for practitioners in several ways. Here are some examples of value-added customer outcomes it can support by coordinating automated tasks into end-to-end processes:



Better deployment control

Orchestrated pipelines of automated tasks make it easier to support processes including progressive releases and automated rollbacks to previous release versions. This increases reliability for the customer.



Contextual intelligence

Ingesting data across all tools, workflows, apps, and cloud environments enables powerful intelligence to optimize software delivery effectiveness.



Progressive delivery

An orchestrated release with full visibility enables release managers to target specific user groups through feature flags. Depending on the impact and criteria set, the feature can be further deployed or rolled back automatically.



Accurate software bill of materials

Supply chain security and intellectual property issues are an increasing concern for customers. Release orchestration helps to coordinate automated tasks that maintain an accurate software bill of materials every time a product ships to production.



Real-time evidence collection

Enables absolute clarity on the status of a release at any point in time, while collecting the necessary evidence needed to generate audit reports.

CloudBees Unify: Streamlined Orchestration for Scalable Software Delivery



CloudBees Release Orchestration brings consistency, control, and visibility to modern software delivery. By coordinating tools, pipelines, and environments across the entire release lifecycle, Unify empowers platform engineering teams to scale delivery without the chaos.

Release orchestration enables organizations to automate complex, multi-team release workflows using a visual, repeatable approach—without replacing their existing DevOps tools. Integrations, approvals, deployments, and audits are centralized into a unified control plane, so engineering leaders gain real-time insight and governance from commit to production.

CloudBees Release Orchestration is part of CloudBees Unify, a DevOps solution that also includes CI/CD, feature flags, analytics, compliance automation, and testing intelligence. Together, these capabilities help enterprises transform software delivery into a business accelerator—enabling secure, reliable releases at scale.

To learn more, talk to a release orchestration expert.

Learn more cloudbees.com/get-started



CloudBees, Inc.
cloudbees.com
info@cloudbees.com

Jenkins® is a registered trademark of LF Charities Inc.
Read more about Jenkins at: cloudbees.com/jenkins/about

© CloudBees, Inc., CloudBees® and the Infinity® logo are registered trademarks of CloudBees, Inc. in the United States and may be registered in other countries. Other products or brand names may be trademarks or registered trademarks of CloudBees, Inc. or their respective holders.