WORKFLOW ORCHESTRATION VS. CONTINUOUS INTEGRATION: WHAT'S THE DIFFERENCE?

Workflow orchestration and continuous integration are both automation tools, but they operate at fundamentally different phases of the Software Development Life Cycle (SDLC). Continuous integration tools like Jenkins automate the Build phase—managing code testing and software delivery. Workflow orchestration tools like Control-M take over in production, managing and monitoring the business application workflows that keep critical operations running.

For most companies, Digital Transformation is no longer just a buzzword but a very real shift in how every aspect of the organization operates. That is certainly true for IT, and one of the somewhat ironic characteristics of this shift is a huge emphasis on automation—ironic, because you would have thought that Information "Technology" was always about automation, which is pretty much a synonym for computing. Well, better late than never, I guess.

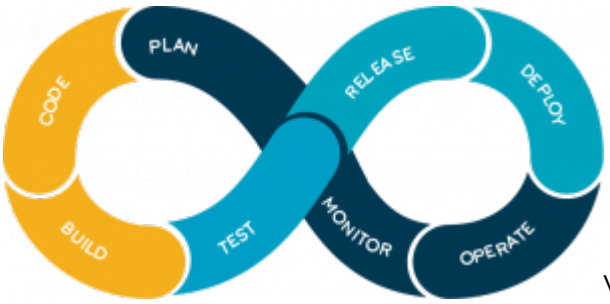
One result of this renewed focus on automation is increased scrutiny of tooling: a desire to understand the similarities and differences among the vast array of available tools and which tool is best for which function. To make that comparison concrete, this post uses two proxies—Control-M, the leading[1] application workflow orchestration tool, and Jenkins, the most popular continuous integration (CI) tool.

What is the difference between workflow orchestration and continuous integration?

The core distinction maps directly to SDLC phases. Jenkins and continuous integration tools are

designed for the Build phase—automating code testing, merging, and delivery. Control-M and workflow orchestration tools operate in the Operate and Monitor phases, providing visibility and management of the production workflows that execute critical business services: customer billing, inventory management, recommendation model training, and business analytics.

The Software Development Life Cycle includes Build, Deploy, Operate, and Monitor phases. When software is still being developed and tested, CI tools like Jenkins are the right instrument. Once an application reaches production and needs to run reliably at scale, workflow orchestration takes over.



When choosing between these tool categories, identify what you need to automate, who will create and operate the automation, and what organizational impact your choice will have.

What does Control-M do?

Control-M focuses on operating business applications in production. According to its own home page, Control-M "simplifies application workflow orchestration, making it easy to define, schedule, manage and monitor application workflows, ensuring visibility and reliability, and improving SLAs." The primary Control-M use case is keeping production business processes running—not building the software that powers them.

What does Jenkins do?

Jenkins is purpose-built for software development and delivery. The jenkins.io site states its mission directly: "Build great things at any scale." The Press page stresses the focus on building software: "Jenkins is ... supported by ... people interested in continuous integration, continuous delivery and modern software delivery practices. Built on the Java Virtual Machine (JVM), it provides more than 1,500 plugins that extend Jenkins to automate with practically any technology software delivery teams use."

What are users saying?

Real-world experience clarifies where each tool delivers value.

Jenkins in practice

T-Mobile uses Jenkins to support customers by "adopting robust and intelligent practices that speed up the CI/CD cycle." Morningstar uses the Jenkins Platform to "improve consistency and increase automation—vital steps along the organization's path to continuous delivery (CD) and a DevOps culture."

Control-M in practice

Todd Lightner of The Hershey Company describes Control-M's production role in this blog post: the data center operations group runs thousands of jobs each day managing digital interactions across manufacturing, supply chain, warehousing, finance, payroll, HR, and sales—all within a complex SAP® environment. As Lightner puts it, Control-M "literally runs our business."

A case study of Raymond James Financial describes their use: Control-M manages jobs across complex interdependencies among hundreds of applications that access the company's data warehouse. Nightly processing powered by Control-M ensures senior management and financial advisors have the data they need to serve clients.

What do analysts say?

Analyst coverage of Jenkins has been relatively sparse, possibly due to its independent open-source status—though an IDC Innovators report covers the space. Control-M has historically been evaluated under the Workload Automation category by Gartner and EMA. The most recent analysis is an EMA Radar Report.

Frequently asked questions

Can workflow orchestration and continuous integration tools be used together?

Yes—workflow orchestration and CI tools are not mutually exclusive. Jenkins manages the Build and Deliver phases during software development; Control-M manages production workflows once applications are deployed. Many enterprises use both tools in tandem across different parts of the SDLC.

What SDLC phase does continuous integration cover?

Continuous integration covers the Build phase of the Software Development Life Cycle. CI tools like Jenkins automate code testing, merging, and delivery to help development teams catch defects early and release software more reliably.

What SDLC phase does workflow orchestration cover?

Workflow orchestration tools like Control-M are designed for the Operate and Monitor phases. They schedule, manage, and monitor the production workflows—batch jobs, data pipelines, and business processes—that keep critical systems running after software has been deployed.

What is the difference between Control-M and Jenkins?

Control-M is an application workflow orchestration platform designed for production operations: scheduling, monitoring, and managing business application workflows at enterprise scale. Jenkins is a CI/CD tool designed for software development: building, testing, and delivering code. Both tools automate processes, but at different points in the SDLC.

Which tool should I choose for DevOps automation?

The right choice depends on what stage of the SDLC you need to automate. For software build and delivery pipelines, Jenkins or a comparable CI tool is the natural fit. For managing the production workflows that run business operations after software is deployed, workflow

orchestration tools like Control-M are the appropriate solution.

[1] Enterprise Management Associates (EMA) Radar Report for Workload Automation

The views and opinions expressed in this post are those of the author and do not necessarily reflect the official position of BMC.