



Continuous Delivery in Java

By Daniel Bryant and Abraham Marin-Pérez

482 pages

Continuous Delivery in Java

Reviewed by Andrew Binstock

October 4, 2019

The 2010 book *Continuous Delivery* by Jez Humble and Dave Farley redefined how organizations delivered software. Until that time, the state of the art was continuous integration (CI), which defined only the automation of source integration and testing. CD began to emerge as the next step. Initially, some viewed it as meaning *continuous delivery*, others as *continuous deployment*, and the vast majority were hard-pressed to tell you the difference between the two. But Humble and Farley's book sealed the definition of CD by describing the many behaviors that continuous delivery comprised. The book was full of useful explanations and practical advice, and it defined the shape of DevOps for nearly a decade.

[Download a PDF of this article](#)

However, much has changed with the advent of containers and the cloud, especially regarding CD and more especially regarding CD with Java. The original volume no longer has much to say to today's organizations. Its fundamental recommendations have been universally adopted and the application of the overarching ideas for containers and the cloud are missing entirely.

Into this gap steps the present volume, *Continuous Delivery in Java*, by the well-regarded Java Champion Daniel Bryant and Abraham Marin-Pérez. Whereas the original *Continuous Delivery* volume focused on the exposition of practices, Bryant and Pérez blend a description of the new practices with a wide-ranging examination of the tools that implement those practices. As such, it has sections on the topics you'd expect—source control, building, packaging, testing, and deploying—but always catering to the present world of containers, VMs, and clouds and viewing them through the issues unique to Java (include the JVM? prefer JAR/WAR files to containers? and so forth). It has additional sections that drill deeper into the primary techniques. For example, the coverage of testing includes sections—sometimes entire chapters—on unit testing, functional testing (correctness and acceptance testing), integration testing, and validation of nonfunctional requirements. The topic of release management includes a more than 50-page chapter on deploying and releasing directly from the pipeline. And postdeployment CD is not overlooked: Monitoring, logging, and tracing are all covered well.

The authors thoughtfully explain cloud-specific matters: the various tiers of cloud services and how to design the pipeline to accommodate them. The two clouds explicitly discussed are Amazon Web Services (AWS) and Microsoft Azure.

Scattered throughout the discussions are extensive forays into the tools that can execute individual tasks inside the CD pipeline. The authors begin with an overview of code management tools (with a special emphasis on Git, naturally) and then many build tools—the big three (Ant, Maven, Gradle) and also the new tools coming up (such as Bazel, Pants, and non-Java build tools such as Leiningen). And they continue for every segment of the pipeline, right through deployment and monitoring. The recommendations are reasonable and well-reasoned. They extend even to replacing tools that were once mainstays of CD but have since been



Sponsored Content

retired or abandoned. (For example, the authors suggest a replacement for the long-loved but entirely abandoned JDepend report.)

The tool coverage is reminiscent of the brilliant *Java Power Tools* from John Ferguson Smart, which is now terribly outdated. But, whereas the latter book had both breadth and depth, the current volume lacks depth in tool coverage. This is a shame, because there are few sources of useful opinions from experts about tools. The authors' choice is probably due to a lack of space, which is a constraint frequently invoked in the book to cut off deeper discussion of tools. But I would then question their inclusion of more than 20 pages introducing the Bash shell. Surely readers can find that information for themselves. Apropos of Bash, the book presents information primarily in terms of Linux, but has frequent detours to accommodate Windows-based readers.

All organizations implementing CD for Java deliverables should have a copy of this book. The explanation of techniques and the many related recommendations (how to implement the techniques and what tools to choose) are clear and practical. Both organizations and individual developers will see immediate and substantial benefits for the time (and money) they invest. Very definitely recommended.

Contact

US Sales: +1.800.633.0738
Global Contacts
Support Directory
Subscribe to Emails

About Us

Careers
Communities
Company Information
Social Responsibility Emails

Downloads and Trials

Java for Developers
Java Runtime Download
Software Downloads
Try Oracle Cloud

News and Events

Acquisitions
Blogs
Events
Newsroom

ORACLE

Integrated Cloud
Applications & Platform Services

