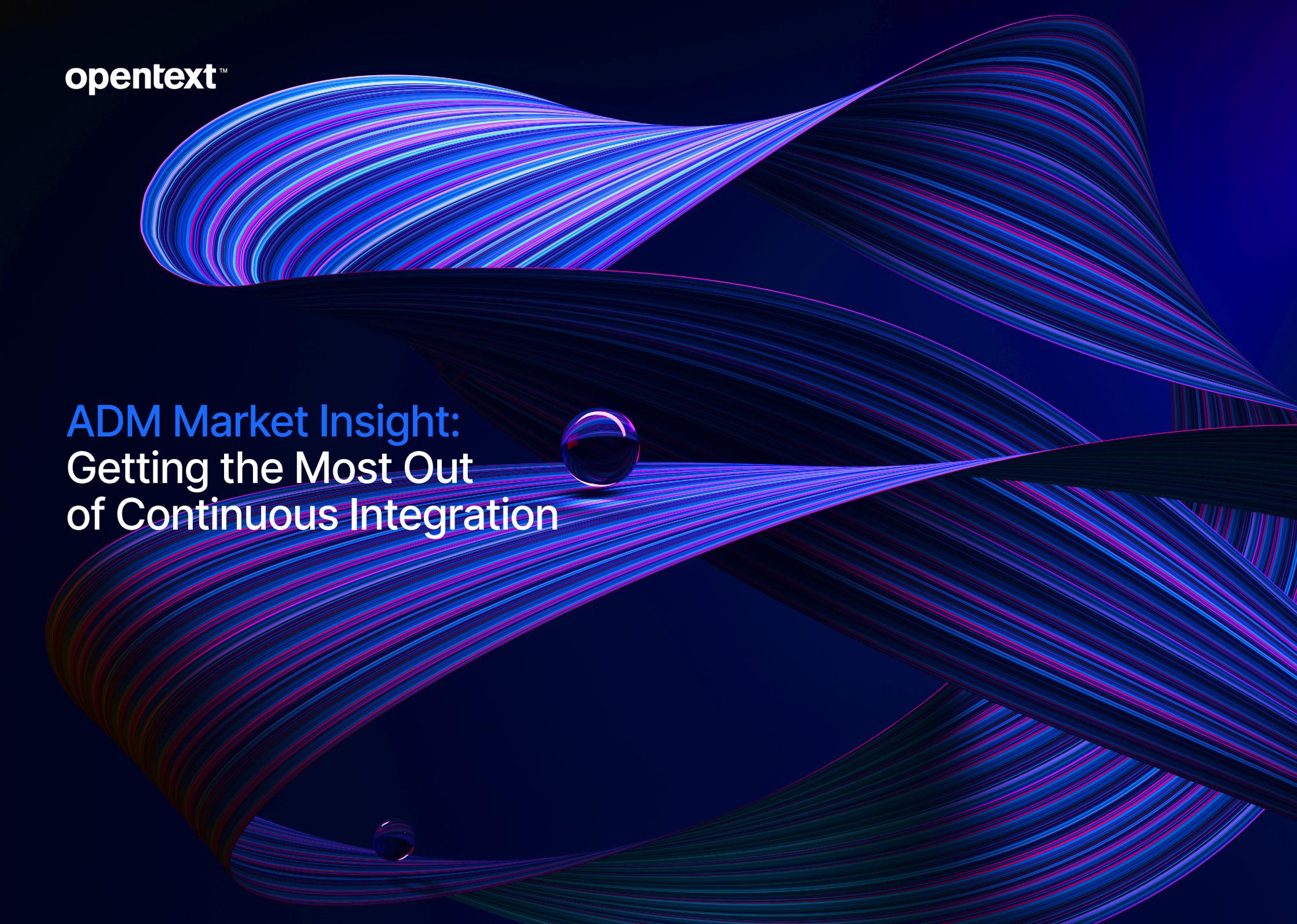


opentext™

ADM Market Insight:
Getting the Most Out
of Continuous Integration

The background features a dark blue gradient with several thick, wavy, ribbon-like shapes in shades of blue and purple. These shapes flow across the frame, creating a sense of motion and depth. Two small, reflective spheres are positioned on the lower part of the wavy lines, one near the center and one towards the bottom left.

Key Takeaways

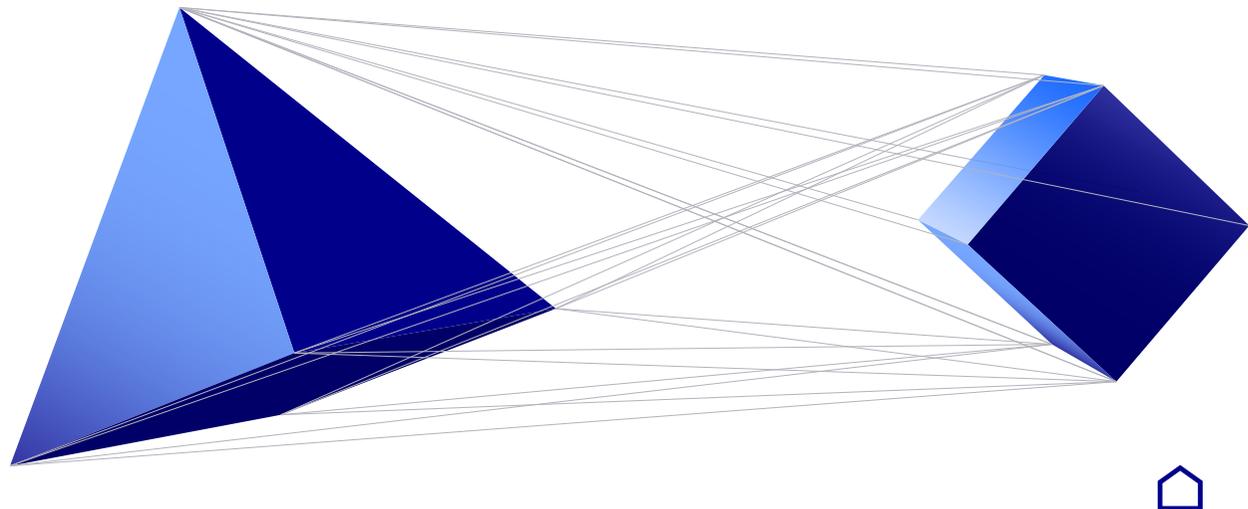
Software teams have made many changes over the past several years. Teams have moved from the traditional waterfall development process to Agile and on-site hardware to cloud-based systems. Amid these changes, a new generation of visual tools has arisen to support the current state of software development.

The Evolution of Software Development

The traditional software development environment consists of a development team; a QA team; and a build, configuration, and release team. Development creates the code and hands it over to QA for testing. QA performs various automated and manual tests on the software and files bug reports—which they then send back to development. Eventually, QA signs off on the software and delivers it to the build team to create a release.

The advent of Agile approaches cut development work into manageable chunks. Teams can integrate development with testing to produce more code in a shorter time. Agile teams often divide work into sprints: short periods of time, usually no more than four weeks each. Every sprint has a specific, designated development goal. Throughout the process, teams assess work in meetings called Scrums.

Cloud computing environments, such as Amazon Web Services and Microsoft Azure, are perfectly suited to support Agile development models. They produce the ideal infrastructure for the development of the continuous integration/continuous delivery (CI/CD) development process.



Continuous Integration/ Continuous Delivery

The idea of CI is that changes to a software environment are built on a frequent basis—generally upon each code commit. Teams deliver changes to a shared repository in a manner that reflects the ongoing effort of development teams. Closely tied to CI is CD, which refers to an automatic pipeline for delivering those changes to a test, staging, or production environment. The two together are often referred to as CI/CD.

While CI/CD is a relatively well-known software development process, the tools for implementing it have developed rapidly. There are several new, innovative tools out there, but many organizations still rely on legacy CI tools. For example, Jenkins is a popular open-source CI/CD tool. It is available on most platforms and supports a variety of extensions and plug-ins. Another tool, TeamCity, is Java-based and runs on Windows and Linux servers.

As useful as these scripting tools are, they have several disadvantages. They are:

- **Hard to learn.**
- **Take excessive time and effort to configure.**
- **Are team-centric in nature.**
- **Provide little visibility into the ongoing delivery process for the business side of the organization.**



A New Generation of CI/CD Tools

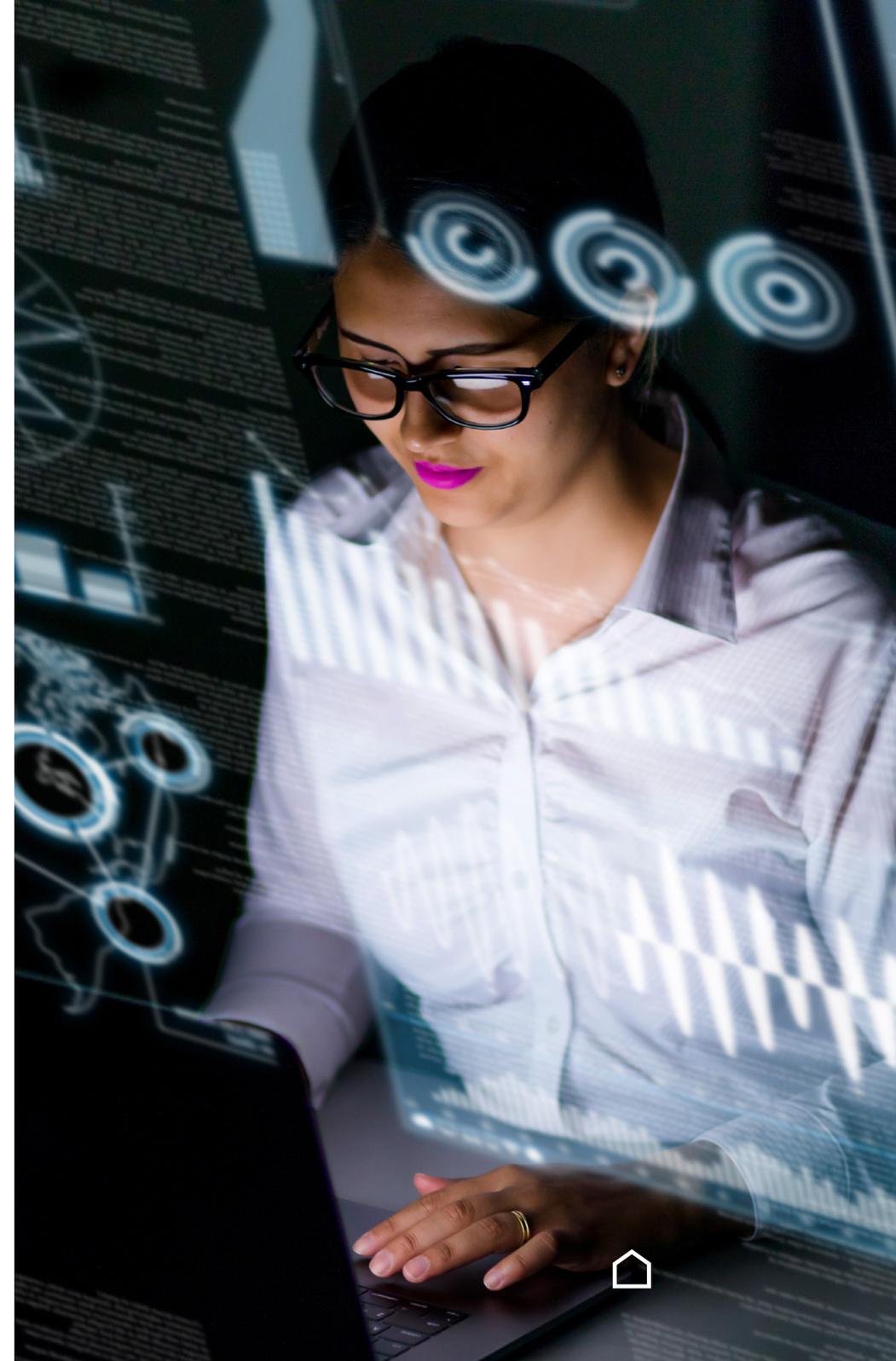
A new generation of CI/CD tools streamlines the process by offering quick and easy configuration using a visual interface. Drag-and-drop configuration simplifies making changes. And a graphical interface makes CI/CD flows easy for everyone to understand, including marketing and business teams. Being able to track your change from inception to delivery—even when it's in the development “black box”—is key to understanding how it's progressing.

These modern tools facilitate quality in the CI/CD process. The end-to-end visibility into the build and deployment process provides accessibility for technical and business users. It enables them to easily spot and fix problems early. These tools help organizations understand how far through the delivery pipeline potentially valuable business assets have progressed.

Today's tools provide an integrated development, security, and operations (DevSecOps) environment. By integrating artifact management and vulnerability scanning into the CI pipeline, you can identify security issues early and prevent them from reaching the production environment.

A graphical approach to CI means that stakeholders at all levels of the business can understand what's happening. The software lifecycle no longer seems like a black box.

Modern CI/CD tools are easily scalable, and manual server configuration is not necessary. You do all the configuration using the graphical interface of the software. The best tools provide you with a “single pane of glass” that allows you to effortlessly manage your application lifecycle and provides useful data to both the technical and the business sides of the company. In short, modern CI/CD tools provide insight, control, flexibility, and scalability via an easy-to-understand user experience that both technical and non-technical users can access.



Modern Workflow and Deployment

When you work in a CI/CD environment, how you manage workflow or pipeline is critical. You should be able to monitor progress and manage quality agnostically in your systems and throughout your application lifecycle.

Today's tools provide comprehensive DevOps tool-chain management and reduce integration costs by managing DevOps with a delivery pipeline connected to IDEs and CI/CD systems. You can implement REST APIs and integrate open-source tools, such as Git and Ansible. These capabilities allow for a governance framework that supports your business rules and effectively manages user roles and permissions.

Because of the frequency of change in a CI/CD environment, quality management is essential. Current tools support multiple testing types and frameworks that enable you to develop both manual and automated testing routines. They ensure quality and accelerate Agile development and delivery by:

- **Identifying the root cause of failures.**
- **Tracking code commitments.**
- **Performing tests and analyses to identify risks early in the delivery cycle.**

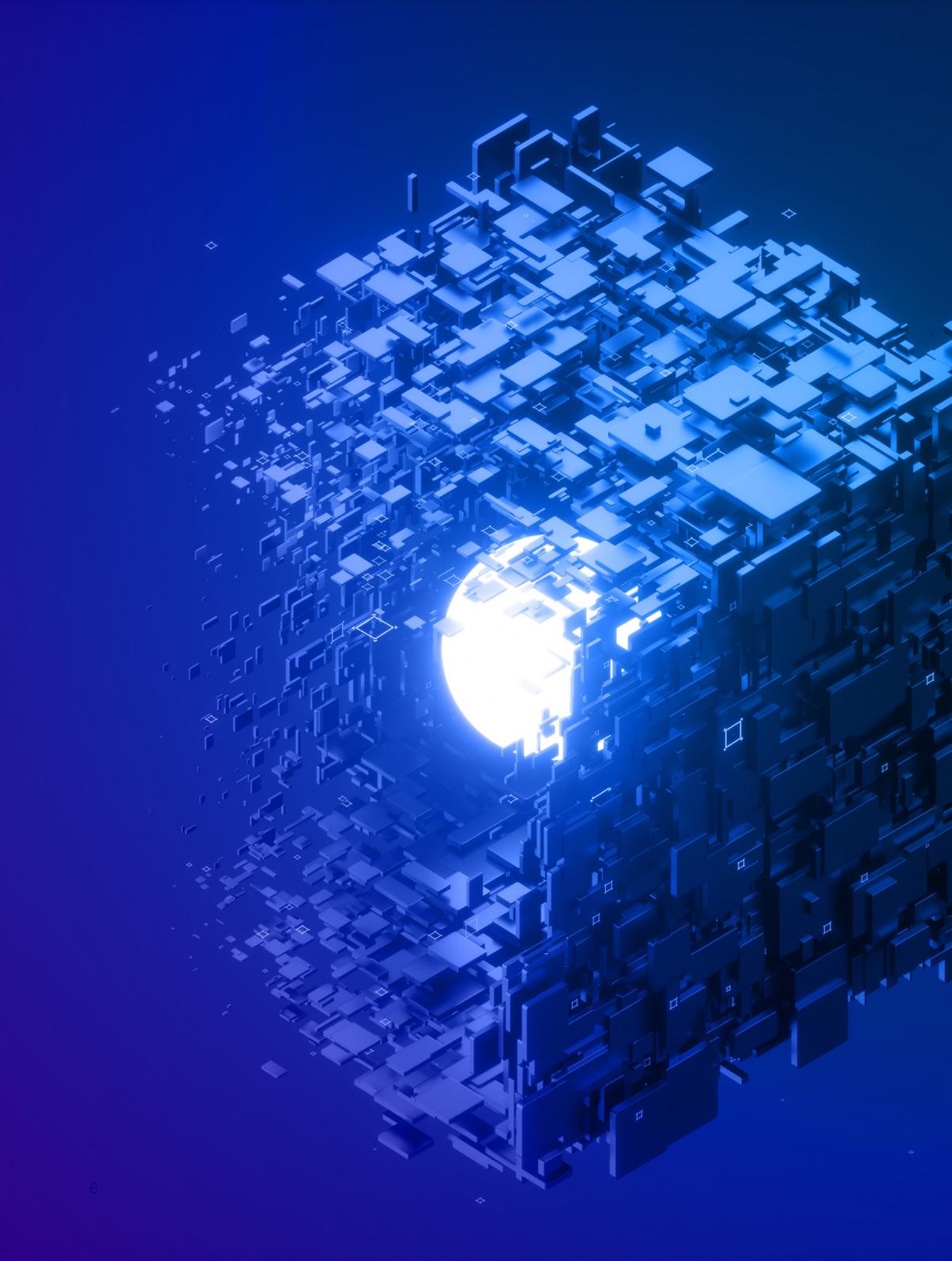
Another important aspect of being successful in a CI/CD environment is Agile team collaboration. Modern tools enable you to plan and develop projects, execute tasks, and accelerate delivery. They also help you scale Agile processes beyond the team level by supporting thousands of users and third-party frameworks. Real-time insight into the status of application delivery, from an Agile project management perspective, and the CI/CD execution status is valuable. Businesses can fully appreciate the ongoing status of projects or product delivery cycles.

At OpenText, we offer modern CI/CD tools. OpenText ValueEdge and Deployment Automation enable you to create graphical build and deployment processes that can be templated and reused across applications. The simplified, intuitive navigation model means that even novice users can become productive immediately. ValueEdge supports full GitOps models with integrated Git server-based code management, continuous integration, vulnerability scanning, and a process-centric artifact vault. Deployment Automation is highly collaborative and supports rapid, enterprise-wide adoption among release teams and tracks the flow of product delivery.



With ValueEdge, you can easily model and manage deployment pipelines at both the application and component levels. Using snapshots to represent specific states of an application, Deployment Automation also supports single-click promotion or fallback through the deployment pipeline using repeatable, audit-compliant models.





Conclusion

Development teams often struggle with the CI/CD lifecycle. Hand coding deployment scripts and routines can be tedious, time-consuming, and error-prone. Such processes can slow down the entire CI/CD process.

It's time for a modern CI/CD environment. You've adopted Agile and have moved to the cloud. Now it's time to bring your CI/CD into the 21st century as well. [ValueEdge](#), can help you achieve this.

ValueEdge Agile enables you to collect pipeline run and automated test run results from the CI server. You can configure both test properties and environment data on the pipeline steps that run your automated tests. Easily track and analyze build results and include automated test run results in product and release quality analysis. Moreover, you can easily track security issues when you discover them in your code.

ValueEdge provides a powerful graphical CI environment with support for multiple SCM tools, in-built artifact management, code review, vulnerability scanning, and remote build support. With native integration to ALM Octane and deployment automation, ValueEdge provides the CI glue to modern CI-based workflow.

The deployment automation features integrate smoothly with existing CI/CD tools. It provides you with an extensive library of plug-ins that integrate with your entire CD pipeline and replaces homegrown scripts and manual processes. With its highly extensible architecture, you can configure the plug-ins during design or at runtime. You can support virtually any system, physical, mainframe, virtual, cloud, or container-based.

[Learn more about ValueEdge and get a free trial to experience the benefits fro yourself.](#)

[Learn more about ValueEdge](#)



opentext™

