



Modern Application Development: Evolve Your Development Processes

Developers drive modern business



Application development has become core to business success. As a result, application developers have emerged as precious assets not just inside their companies, but across the global economy. According a report from Stripe, these developers have the potential to collectively raise worldwide GDP by \$3 trillion over the next decade—which is why keeping them productive and professionally fulfilled is more important than ever.

Attracting & Retaining Talented Developers

Many developers want to work in an environment where they can focus on creating features that deliver value. To foster this kind of environment, many companies are reorganizing, uniting their development and operations groups to form DevOps teams that have the autonomy to build, run, and own their service.

These teams release features using new processes, such as Continuous Integration and Continuous Delivery (CI/CD), which accelerate software delivery.

Modern Software Delivery with CI/CD

CI/CD is a process that involves automating the build, test, and deploy phases of software development to achieve faster releases, higher quality code, and less risky deployments.

Automating tasks in the software release process improves developer productivity, while significantly reducing error rates. These practices optimize the day-to-day experiences of your developers, and free them to focus on innovation that benefits your business.

“Setting up that strong developer culture is important for attracting and retaining talented people. Moving to DevOps on the cloud is just another way that we can cultivate and support independent, autonomous teams that feel empowered to do their best work every day.”

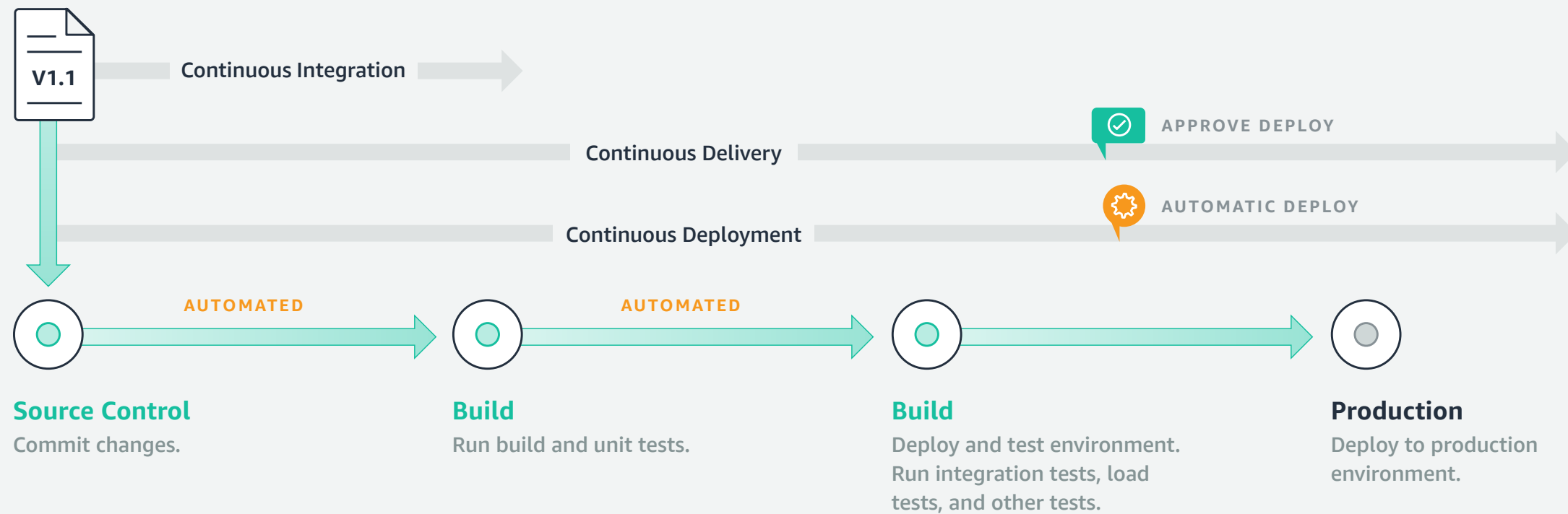
George Brady, Executive Vice President and Chief Technology Officer at Capital One



Modern Application Development: Evolve Your Development Processes

Automation, from build to production

CI/CD automates the entire software release process. Every revision that is committed triggers an automated flow that builds, tests, and then deploys the update.





Modern Application Development: Evolve Your Development Processes

Best Practices for CI/CD

- ✔ Treat your infrastructure and delivery configuration as code
- ✔ Keep DevOps teams small (limit to 12 self-sustaining members)
- ✔ Standardize builds on systems
- ✔ Assemble unit tests toward 100% coverage of the code base
- ✔ Make unit tests 70% of the testing in duration, number, and scope
- ✔ Establish role-based security controls
- ✔ Capture, track, and measure standard metrics
- ✔ Commit code to the main trunk, not long-running feature branches
- ✔ Use distinct pipelines for each branch and team

Benefits of using CI/CD

Attract and retain top talent	Improve developer productivity	Find and address bugs quicker	Deliver updates faster
Reducing the time developers spend on the tasks they like least, and freeing them to focus on innovation fosters a culture that keeps talented developers happy.	Automation reduces the time developers spend on routine tasks, and eliminates waiting periods between stages of development.	Your team can discover and address bugs earlier (before they grow into larger problems later) with smaller changes and more frequent testing.	CI/CD provides a deployment-ready build artifact passed through a standardized test process that can be automatically deployed into production safely and quickly.

Happy developers, happy customers

By implementing CI/CD processes, developers have the framework to more rapidly turn innovative ideas into a practical reality. This also fosters a developer friendly culture, freeing time for your development teams to focus on what they love: building great products. Evolving your development processes with CI/CD is one of the best ways you can attract and retain the most talented developers, while also building and quickly releasing features that delight your customers.

Visit us to learn more about how to leverage CI/CD in your development environment.