



### Industry:

Financial Services

### Geography:

Ireland

### Summary:

One of Ireland's "big four" commercial banks improved the efficiency and cost-effectiveness of software development operations with DevOps powered by CloudBees.

### Challenge:

Modernize software development to reduce costs and outcompete challenger banks

### Solution:

Adopted CloudBees CI to support implementation of CI/CD process automation and manage pipelines more efficiently.

### Results:

- » Up to 30% reduction in development cycles projected
- » Reduced over 1,000 CI/CD scripts to about 300
- » Completely eliminated open source Jenkins instances
- » Supported automatic deployment of microservices

### Product:

- » CloudBees CI

## 'Big Four' Irish Bank Stays on Top of Industry With DevOps

One of Ireland's "big four" commercial banks offers a full range of personal and corporate banking services as well as insurance products and pensions through Irish Life Assurance plc. The bank is almost wholly owned by the Irish government.

The banking industry isn't what it used to be. Mobile and internet banking has taken off as more customers go online for services instead of walking into brick-and-mortar branches. Banks are now competing on the basis of new digital features and capabilities that provide a better experience to a new generation of tech-native customers.

The trend is placing a new burden on financial institutions. To stay ahead, banks like this one are stepping up investments in modern digital platforms and devoting more resources to software development and infrastructure. As the head of DevOps CI-CD & Enterprise Dev Tooling wryly observed, "We're not a bank that writes software anymore. We're a software company that just happens to be a bank."

---

*"CloudBees CI enabled us to have shared masters, but more importantly, to standardize on development patterns and shared pipelines. Now we could go to the whole bank and say, 'This is the official pattern you must follow for every new project.'"*

---

Head of DevOps CI-CD and Enterprise Dev Tooling

## Challenge

### Modernize Software Development to Cut Costs and Outcompete Rivals

Seeing the writing on the wall, the bank launched a major digital transformation a couple years ago. The initiative included a significant push to modernize how the bank developed and deployed software, with the aim of speeding software release cycles, reducing risk and curbing costs.

Even before the transformation, engineering teams had started embracing new agile methods and tools to automate the development lifecycle, such as DevOps and continuous integration and delivery (CI/CD). Many teams had adopted open-source Jenkins to automate pipelines, but these efforts were uncoordinated and splintered, slowing progress.

“Our software development tools were scattered around the IT organization,” he says. “We had no central strategy for tooling. There were too many people doing it their own way.”

The disjointed approach was also costly. “We were hiring consultants to reinvent the wheel for every development team, because every team believed they were different,” he says. In reality, though, most of the processes were similar and could be standardized and shared across all the teams.

As he explains, “If we really want to do CI/CD and we want to standardize, we don’t want to see all these Jenkins instances all over the place that are hard to manage.”

The idea of standardizing processes and cutting costs appealed to bank executives who were keen to avoid risk and secure predictable business outcomes. “Every bank’s nightmare is to appear in newspaper headlines – for the wrong reason,” he says. Similarly, faulty software that triggered a service outage could turn off customers. Another ongoing concern: the threat of fines by regulators who oversee the bank’s core banking processes. Overall, he says, “We needed to be sure that we got everything right with our software processes and that our releases are approved and auditable going into production.”

## Solution

### Adopt a Centralized CI/CD Platform With CloudBees

One of the head of DevOps’s first moves was to build a business case for standardizing on a common DevOps platform using CloudBees CI, a CI/CD automation engine that supports diverse software portfolios and unified governance. Faster development and speed to market was a key advantage of moving to CloudBees: he estimated that the organization could reduce cycle times by about 10% and possibly up to 30%. On top of that, the business case projected a big boost in efficiency from streamlining the administration of masters, saving the equivalent of about a half person per server.

Impressed by the potential advantages, executives green-lighted an investment in CloudBees CI, and his team got to work centralizing the organization’s software development tooling. “If you really want to get automation, you need to get your tooling in order,” he says.

To standardize on CloudBees CI, the head of DevOps collected scripts from each development team. At the time, teams were using about 1,000 CI/CD scripts – most of them sharing a similar structure. These were narrowed down to about 300. “We got rid of duplications in Jenkins and migrated them into proper pipelines in CloudBees CI,” he says. “CloudBees CI enabled us to have shared masters, but more importantly, to standardize on development patterns and shared pipelines. Now we could go to the whole bank and say, ‘This is the official pattern you must follow for every new project.’”

The bank is reaping the benefits from standardizing and centralizing its DevOps environment, including a marked increase in developer productivity. In one case, a developer hired to work half time produced as much code in three to four months as a typical full-time developer would produce in a year and a half. The head of DevOps credits the boost to efficient strategies and patterns available on the CloudBees CI platform, along with access to shared code. Instead of wasting time “reinventing the wheel,” he says, the developer was “out there producing value.”

The shift to lean and agile methods underpins what he calls a “brand new way of working” for the company’s development teams. “It’s about taking a lot of the principles from lean and DevOps about cooperation, trust and transparency – and bringing them across the whole organization and revamping all our processes,” he says.

Today, the entire organization has moved from open source Jenkins to CloudBees CI pipelines. “Nobody’s using free-standing Jenkins or Jenkins scripts anymore,” he says. The change has significantly reduced complexity and costs. “If we’d have kept going with open source Jenkins, we would’ve had at least 20 if not 40 instances of Jenkins servers out there,” he says.

---

*“Instead of wasting time reinventing the wheel, the developer was out there producing value.”*

Head of DevOps CI-CD and  
Enterprise Dev Tooling

---

The bank originally had 20 servers that had to be maintained but now has 10 virtual machines that can be shared. “We had 20 people handling the administration of all those services and now we have just one,” he said. “Now, those other employees can work on more high-value tasks.”

The bank also tapped into the expertise of CloudBees professional services to set up its pipelines and to manage the security requirements of the bank. “There were 220 commands that security needed addressed. CloudBees Professional Services got engaged to each and every one of those commands for our security team.”

For teams focused on building microservices – a lightweight and increasingly popular software architecture – this bank’s “new way of working” using CloudBees-driven patterns and pipelines has become all but mandatory. “We insist they use a shared microservices pipeline,” he says. “We do all our new work on that pipeline. It really has helped us standardize.”

What’s more, the bank is leveraging CloudBees CI to automatically deploy new microservices into production. However, when it comes to legacy applications, the head of DevOps is taking a more conservative approach, noting the higher level of risk – and audit requirements – associated with releasing more traditional software products. Still, he hopes to solve these challenges soon enough, allowing the bank to “automatically and seamlessly do continuous delivery into production” for the vast majority of its applications.

Supported by an efficient, centralized, cost-effective software development platform, the bank’s digital transformation is empowering the bank to stay on top of a fast-evolving and increasingly digital financial services industry.

## Results

### Faster Time to Market.

“We reduced cycle time by about 10 percent and can see that increasing to 30 percent,” the Head of DevOps CI-CD and Enterprise Dev Tooling says.

### Greater Efficiency.

“If you really want to get automation, you need to get your tooling in order,” he says.

“We got rid of duplications in Jenkins and migrated them into proper pipelines in CloudBees CI,” he says.

“CloudBees CI enabled us to have shared masters, but more importantly, to standardize on development patterns and shared pipelines. Now we could go to the bank and say, ‘This is the official pattern you must follow for every new project,’” he says.

### More Cost Effective.

“If we’d have kept going with open source Jenkins, we would’ve had at least 20 if not 40 instances of Jenkins servers out there,” he says.

### Lean and Agile.

“It’s about taking a lot of the principles from lean and DevOps about cooperation, trust, and transparency – and bringing them across the whole organization and revamping all our processes,” he says.

### Greater Developer Productivity.

“Instead of wasting time reinventing the wheel, the developer was out there producing value,” he says.

---

*“We reduced cycle time by about 10 percent and can see that increasing to 30 percent.”*

Head of DevOps CI-CD and  
Enterprise Dev Tooling

---