CloudBees.

Case Study

SOMOS™

Industry:
Telecommunications

Geography:
North America

> *"Before CloudBees Flow, our deployments used to be agonizing and take anywhere up to three days. Now, these same deployments take less than two hours."*

Gary McKay
Director, Agile Service Delivery
Somos

Want to Add a Smile to Your Developers' Faces? Make Your Deployments Boring

Contributed by Gary McKay, Director, Agile Service Delivery, Somos

At Somos, my DevOps teams aren't simply process developers, we're DevOps evangelists.

While we may represent more of the "Ops" side of the business, we are DevOps in our principles and operations world view. We provide the core tools, capabilities and expertise needed to support CI/CD processes at the team and department levels. We take pride in setting up the environments and establishing pipelines, whether they are containerized applications, more traditional or support a large enterprise application like our SMS/800 Toll-Free Number (TFN) Registry.

Somos can be likened to a DNS for Toll-Free Numbers in North America, in that every time you dial a Toll-Free Number, Somos is the number's registrar. We make sure that the registry is a fair and neutral playing ground to prevent one large telecom from taking control of another telecoms' number without following proper protocol. Our customers are not the end user; however, but rather the Responsible Organizations (Resp Orgs) that supply Toll-Free Numbers to the businesses.

Who are my end users in that case? The engineers within Somos. It is my responsibility to ensure that they are able to focus on their core competency — development — and don't get bogged down by the pipeline or the development environment. I even have one developer on the engineering team that serves as my go-to litmus test. Whenever I see her smile because she's happy on a project, I know that I've done my job. That's where I provide a key value-add: I take care of the process-ridden minutia so that she and her peers can focus on what they do best—being awesome developers and solutions delivery rock stars. In my role, I work diligently to eliminate onerous pain points so that our people can successfully deliver innovative solutions to our business.

Getting Away From Lengthy Deployments

Four years ago, our COO identified a major roadblock in our path to innovation — our main system's mainframe infrastructure. The mainframe was aged and outdated, which made it costly and difficult to maintain. It likewise stopped us from adding new features and leveraging innovative technological advancements that could help better solve our customers' business problems. As the migration to our own data center progressed, it became apparent that the transition was having a far greater impact on our deployments than had been anticipated and that the process had become significantly complicated.

Deployments in the new data center were error-prone, not repeatable and often took more than just the day or two they should. To add insult to injury, that time spent only accounted for the deployment itself. Don't get me started on how much extra time was needed to complete the supporting processes, like regression tests and validation. It was all such a slog. It literally took an entire weekend, an army of engineers and all the Starbucks coffee in Seattle to deploy a single release. We're talking upwards of 30 people, both in the United States and offshore, coordinating a Herculean effort every time. As a team, we knew this system was completely unsustainable.

And so my search for a systematic way to deploy software was born.

The only way to sustainably deploy with such a high-level of complexity is to automate as much as possible. In partnership with our superstar team of vendors, we were challenged to come up with a solution that married modern DevOps principles with our 20-year-old legacy architecture.

The criteria list for the new solution was specific and comprehensive, with technologic agnostics sitting high at the top. I didn't want to lock us into a specific technology, particularly since our on-prem data center is VMware, but our DR site is in AWS. It was critical that any solution developed could support either environment. Another priority was that the solution be future proof. Whether there be changes in the application stack, components or environment, the solution should remain unaffected and continue to deploy business-as-usual.

I knew that the tools I was envisioning had to be out there, so my first step was to reach out to my network. I was constantly on the phone those days asking anyone I could get ahold of: "What are you using at your shop?" Luckily, I have some really solid contacts that were able to recommend a slew of tools that were helping them. One such recommendation was for VersionOne. While they were starting to make a splash in the DevOps orchestration space, their tool was a proprietary one and not a fit for what we were looking to accomplish at Somos. It was in my search that I came across CloudBees (formerly Electric Cloud) and got into researching CloudBees Flow.

Investing in a Relationship, Not Just a Product

When it came time for implementation, a CloudBees Flow expert was at our side every step of the way — including installation, build-out and configuration. Knowing that CloudBees Flow is a robust tool overflowing with capabilities, we definitely made it a point to take advantage of their expertise and milk those consultation hours. Similar to any new implementation, there was definitely a heavy lift involved with getting up and running is best executed by having its own dedicated resource. Personally at Somos, we had a handful of experts help us build the application models and act as valued partners to the DevOps team.

We began by gaining an understanding of how we needed to deploy and from there, built the application model around those specs and our own existing architecture. As the complexity of the application grew, so too did the application model. In the beginning, we started with only a couple layers in CloudBees Flow. But, by the time we were done, we wound up with 16 (yes 16!) layers.

Perhaps one of the best parts of CloudBees Flow isn't just the tool itself, but the support I received from CloudBees. Everyone I talked to was helpful and empathetic. They weren't there to simply sell me a tool. They were genuinely invested in ensuring that their application successfully addressed and solved Somos' business needs. While some interactions with companies feel purely transactional, the CloudBees team is one of those rare gems that are truly authentic.

Anyone with a trained sales team can sell me a product. What CloudBees did, however, was to sell me a relationship. Never once did I call "customer support" when I had a question or issue. I called a real life Customer Success Manager who was able to provide the personal, customer-obsessed attention I needed. I call our partnership a relationship too. Here we are, four years later, and CloudBees Flow is still with me, every step of the way.

> *"Perhaps one of the best parts of CloudBees Flow isn't just the tool itself, but the support I received from CloudBees."*

Deploying with Confidence

Before CloudBees Flow, our deployments used to be agonizing and take anywhere up to three days. Now, these same deployments take less than two hours. Likewise, we have been able to drastically cut down on employee hours. Before, we may have had 30 people on the phone for a deployment. Now it takes just five. With each deployment no longer being a mammoth undertaking and huge event, I honestly couldn't tell you the number of daily deployments and builds we do now. In fact, we have it down to such a science that releases are almost (Dare I say it?) predictable and boring. Gone now are the days when we had to devote massive resources for a single deployment and pray that we didn't break the system in the process.

These days, we're completely killing it and knocking out anywhere from 20 to 50 builds a day. It's not just time and efficiency that have increased though, but also our confidence. In fact, we've now gotten to the point where we can no longer do this manually. Partly, it's because the deployments and the environment are so complex that no one should deploy by hand. But this is also attests to how integrated and important CloudBees Flow is to the team. We don't want anyone doing manual deployments.

> *"Using CloudBees Flow has been a true game-changer and resulted in fewer breakages which reduces the load on the help desk, and maximizes customer experience and satisfaction."*

Using CloudBees Flow has been a true game-changer and resulted in fewer breakages which reduces the load on the help desk, and maximizes customer experience and satisfaction. Whereas customers used to get anxious just about every time a new release was announced, these days they are able to sit-back and be rest assured that each roll-out is delivering maximum quality and value. As a bonus, with all the time saved, our developers are freed up and are able to commit their time to developing new and exciting customer requested enhancements. No more 18-month long development cycles. Deployments now can be completed in as little as a few weeks!

Did I also mention weekends? Yes, thanks to CloudBees Flow, gone too are those nasty three-day deployments. That, right there, is a reward in itself!

Some Parting Words

I've learned at least two lessons from this journey. First, it's critical that your architecture be cloud-native, as it immensely helps to drive the CI/CD process. Second, make sure you are always looking at the whole picture. Don't wait until the application is built to try to figure out how to deploy it. Instead, design your CI/CD process with the application it's going to support at the forefront of your mind.

When starting a conversation about what will go into a new application, have DevOps at the table. No ifs, ands or buts. How you execute a successful deploy and release is directly connected to your CI/CD process. So why would you want it to be an afterthought? Instead, make sure critical activities such as automation testing, building your pipelines and ensuring architectural compatibility are built into the process and that consistency is established early on. Regardless of the environment, the process should flow the same way each and every time.

When there are no surprises, there are no headaches. And when there are no headaches, our lives are infinitely easier. This is the gift I like to give our developers — the gift of no surprises. When developers are free to focus on what they need to do — meeting customers' needs, growing the business and helping break into new markets — that's when real success is achieved. Easier deployments, fewer bottlenecks and weekends, yes, weekends. Those are the things that are sure to put a smile on any developer's face.

Learn More About Somos

www.somos.com

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com