



AMBIT

Creating Some Space

When Ward Burson ran CIO at Southwest, he and his team had an ailing array of systems that were costing the airline more than it could afford; in fact, the airline could no longer afford to run them. Particularly problematic was the legacy software that was bogging down the company's ability to open a multiplicity of websites to serve its 100 million or so users by their mobile devices. That job fell to him.

"We all agreed that the first step of the project was to identify the need for reengineering," says Burson. "Many projects are successful with a right out of the box, as we decided to use a 'clean-slate design'." So Burson and his team developed a new architecture of legacy systems and present them to users in a mobile environment.

The CIO's team took away the issue of storage and compute from the IT department, and the team focused on the mobile devices. They began prototyping those devices before they completed their prototype. The final prototypes allowed an array of users, including mobile devices, desktops, and web, to download the application simultaneously. "It was a challenge to make sure that all these different devices were able to work in the individual devices," says Burson. "We paid a lot of attention to the details," says Burson. "We used much better, for example, in giving a range of colors rather than just black and white."

"One of the best parts of mobile technology allows us to do things that we can't do with desktop technology," he says.

The mobile platform will have increased revenue over time when the company begins offering mobile devices, and the additional month

of revenue generation is given these extra space to grow. "It's approach is different to be pursue a mobile business strategy versus a desktop or laptop," says Burson.

For a user, the mobile device is a great way to interact with your flight information and travel plans, and it's a great way to engage with the airline.

With the new mobile application, the airline will be able to offer a better service to its customers and a better experience for the passengers.

Developed through a series of the development cycle, the mobile application has been well received, but not without some hiccups. In other words, the process has a collection of mobile bugs instead of a working CIO.

Breaking Down the Big Application

While Ambit Energy, which provides gas and electricity to deregulated markets, was only just founded in 2006, CIO John Burke, who founded up within his own iceberg to dismantle. At the company's launch, he scoured the market for management software and business, but he didn't find what he was looking for, so he and his team built their own.

When the business started to take off, growing from zero to \$325 million in three years (11 billion total), Burke saw that the company was running almost all of the business—including billing, ratings, customer care, and transaction management—on one huge monolithic system.

"We were growing and expanding into new markets and running the business on this one monolithic system," says Burke. "It was painful to do deployments right; we would set midnight, go all night, and then the morning team would come in and repair what was wrong," he says. "It was hard to pin down a production environment in our test environment because the application was so big and complex. The process burdened our DevOps team and frustrated the business."

Burke recognized that he had to replace the Waterfall development model his team had been using with something better. "Waterfall was such a long road," he says. "We would define some new functionality, but by the time we deployed it, the business was no longer interested and had already moved on to the next thing."

It took a little convincing of both his development team and his business partners, but over the next eight months, Burke invented Ambit Energy's DevOps development model. In another important move, he reorganized his department to have nine dedicated software teams to align to the company's various business units.

"At first, everyone was very excited about what, to us, was a pretty radical organizational change," Burke says. "Next said, 'This is great. The business line managers can go directly to their dedicated development teams and tell them what they want.' It was very empowering to them, and they were motivated to make a lot of change."

But that excitement was short lived when the development teams realized that they had not solved the primary obstacle to driving rapid change. "The development teams were broken into business units, but we still had one large, central system, so we were still doing

late-night pushes," Burke says. "We had a bit of a moment of 'Eh, we've got this movement, but then we realized that nothing changed.'

"When it came to making system changes, we were in the same place."

DevOps at Ambit Energy

Around that time, Burke's senior developers and architects were all talking about Simon Stoyanov and how Amazon, Netflix, and Google were changing their development processes. "People were talking about how fast fail fast, how DevOps gave to his development team at Amazon when they thought they had a 'mission-critical' problem," Burke says.

"Bezos told them, 'You need to take your own piece of the large application and rip it out from the rest. As long as you provide APIs to the large application, you can write your own piece in any language. If your little interfaces have to always be working,'

Says Burke, "So I asked him, 'How do you do this transition?' He talked about whether they could rip their application up." Burke adds, "There was a small minority of developers who thought it was doable, so we let them go. He chose," she says.

Introducing DevOps

Burke's first team renamed the software configuration team "DevOps" in part to herald the direction changes for the organization. "Software configuration was considered 'boring' to get code into production, while 'DevOps' meant automation and pushing quickly code fast. Our DevOps people were not, necessarily, happy when we first renamed the team," says Burke. "But we had to move their manager, from gatekeeping to facilitating change. That was one of the hardest changes, to get them to see their role as facilitating rapid deployment in pushing code."

...which means that, rather than write the code,

developers now had to configure deployments that were automatically correct, and they all could be done by tempo and automated software. "That kind of focus is really hard to think through all of the configurations on your system," says Burke.

Burke identified how key architects will believe in the automated continuous delivery model and coached them to bring the vision to the rest of the team. "The group that influenced these leaders, the people who had been tasked to automate development were starting to get excited about it."

While Burke was reorganizing and appointing leadership, he still had this large application he had to split up, and the company was growing fast. "We didn't have a clear budget or alignment to figure this out while we performed our first deployment," Burke says. "We were all working on the things that matter to the business, and at the same time we had to figure out how to break one large system into many smaller, business-aligned systems."

To break the application down, the team had to figure out which automation tools to buy, and how many extra servers they would need, but to Burke that was the easy part. In addition to changing the mindsets of the team, there were real challenges coming from outside the business. "It was now 2011, and while the business had gotten excited about the nine dedicated development teams, they started to lose faith when we couldn't get any deployments out the door. We had to convince our business partners to take the chance and allow us to rip apart the application."

In the end, after a two-year initiative, Burke and his team did rip apart the application and are now in automated continuous delivery heaven. "We got to the point where we were doing roughly thirty-four deployments a day and we never saw a hiccup. Our business teams have been on fire because they can work with a purpose; they can get things done."

Burke's experience is a testament to the fact that DevOps is more

than an approach to development and infrastructure. It's much more than a new way of life, it involves a massive change in the way all professionals act, mind their own business, and make mistakes.

"You can never make the leap to DevOps," says Ralph Loura. "The dev can never leave, put in the tool, and walk away. If you think for him to leave because they want somebody's eyeballs on the process. It's like being a parent if you are a parent, you never leave your children, you never leave them alone, they never learn that choices have consequences. It's the same with developers. If a developer makes a mistake and someone in operations catches it before the program goes into production, the developer never learns. But with DevOps, the developer sees the ones who get wrong get updated at three in the morning because their code failed."

Due to poor documentation and engineers' resistance to collaboration, developers remain isolated and some do not. "Developers are afraid and some do not