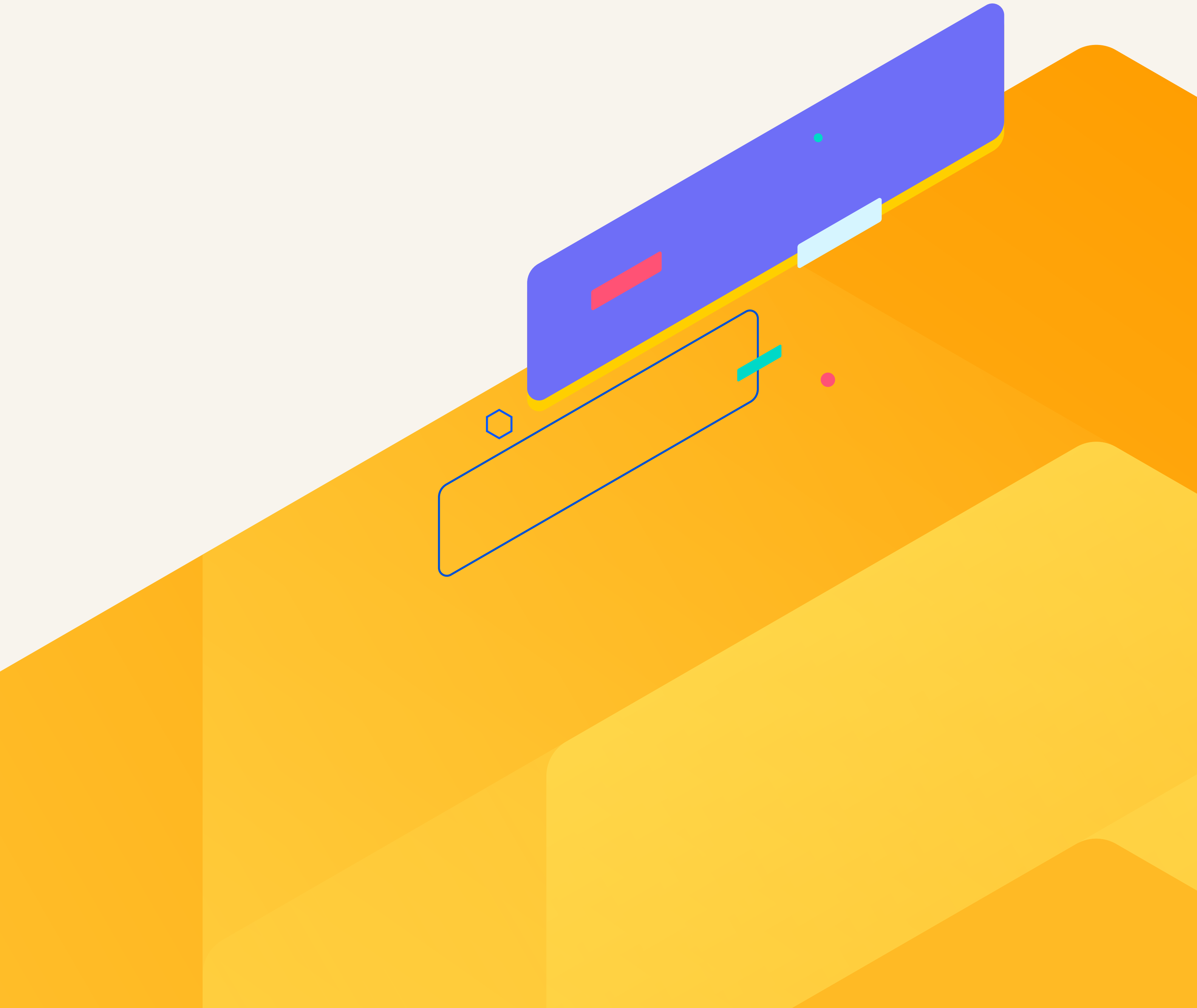
Progress® Chef®

Enabling Developer Services at SAP

CASE STUDY



SAP, the market leader in enterprise application software that helps companies of all sizes and industries run better, has reshaped their processes to enable the speed developers need. In this article, Ran Kahal and Martin Schygulla share insights on their mission to provide developer services at SAP. Ran is director of SAP's DevOps Center of Excellence (CoE); a worldwide team operating in geographically distributed areas like Germany, Israel, France, Mexico, and the United States. His organization creates and manages a variety of test infrastructures where developers can work on their applications. Martin works within that group as DevOps Automation manager for a team that handles Chef, anything else related to automation, and policy enforcement.

As the growth of applications increases year over year, the most successful companies will be those that understand and enable the strategic role developers play in their organizations. In order to enable this success, Enterprise IT organizations must evolve their practices. Groups that provide a classic "Shared Services" model must reshape their goals, processes, and offerings to keep up with this growth. Traditional models focus on implementing risk mitigation and operational standards. A modern "Developer Services" model focuses on empowering and enabling developers to deliver ideas to market quickly.

Using Automation To Manage Complexity

SAP's DevOps CoE began using Chef about two years ago. The CoE manages approximately 10,000 nodes, where about 70% of those machines are Windows servers. The group manages a variety of infrastructure types.

"We can build a simple single server, any type of OS, applications, development tools, and databases," says Ran. "On the other hand, we can also build large, complex networks that we call landscapes. These [landscapes] can include any number of application servers, database servers, and web servers. We do this for hundreds of development teams and each team is different. We build solutions for teams with 10 members and for teams that have 500 members."

> **"We used any scripting language. We used bash, we used PowerShell, we used Python, every script that you can think about or is available. When we were introduced to Chef we slowly started to write the automation and switched to Chef for these deployments"**
>
> **Martin Schygulla, DevOps Automation Manager**

Before Chef, the group used a wide variety of tools. Ran tells us, “We used any scripting language. We used bash, we used PowerShell, we used Python, every script that you can think about or is available. When we were introduced to Chef we slowly started to write the automation and switched to Chef for these deployments.” Chef introduced a common language with cross platform support and the flexibility to automate the most complex of use cases.

Establishing Shared Services

Building reusable automation with Chef meant that the team could then focus their efforts on higher value offerings. The team developed a self-service portal where developers can order the machines or landscapes they need and get them quickly. They call it the Unified Cloud Portal (UCP) and it can deploy machines to both SAP’s established internal cloud and to an external public cloud they’re experimenting with. The machines are then bootstrapped to install the Chef client with all necessary authentication data, initiates a chef-client run, and then Chef takes it from there.

“Users can either choose between an à la carte menu—meaning they can choose an OS, template, and the Chef cookbooks that will deploy automatically on top of this OS and template—or a machine that is ready-made.” Ready-made machines are predefined landscapes containing everything a team might need. Chef’s modular approach to automation means the CoE can shape its offerings to compose the right solutions on demand.

“Before we had Chef and the UCP, it could take developers days to get a landscape. Today, it’s maybe five or six clicks in the portal and then they can go on a coffee break or do something else. The automation can run anywhere from 20 minutes to maybe three hours, depending on the size of the landscape. So it might be a really long coffee break. But the point is that it all happens without the developer needing to do anything.”

Ran’s team has not only changed the way they deliver services to developers but they have also changed their culture. Ran says, “Two years ago, we were a traditional sysadmin team that concentrated on end user task support, server operations, server management, server monitoring and all the SAP tools and applications and databases that are running on top of these servers.” By shifting away from traditional models and embracing automation, the CoE was able to change their focus and reshape their own internal priorities. Ran continues, “we started to speak about DevOps concepts and how to implement DevOps within our organization.”

> **“...it all happens without the developer needing to do anything.”**
>
> **Ran Kahal,** Director of DevOps Center of Excellence

Cooperation Between Development And Operations

The CoE team allows for flexibility beyond use of the UCP so that developers can introduce changes of their own. Martin says, “Developers are allowed to do some software modifications on their systems. For instance, on Linux systems, you can add additional installation repositories where you can look up your RPM packages or whatever. In some cases, developers have root access on Linux systems or admin access on Windows boxes.”

Martin relies on Chef and its test-and-repair approach to ensure that, even with developer modifications, systems still conform to corporate policy. “For operational purposes, we need to ensure that certain security settings and monitoring applications are in place and the developers aren’t just disabling these kinds of things. From my point of view, that’s a huge benefit for Chef. For instance, I don’t need to check if Nagios is installed. I just execute the Nagios cookbook and Chef makes sure that Nagios is there and the right configuration is there.”

Martin relies on Chef to enforce many policies. “For example,” he says, “we have several security guidelines for password strength, the combination of allowed characters, numbers and so on. We don’t need to check if this setting is in place. We force it by applying it in the configuration and we have defined that in the Chef cookbook. We know the machine will have the desired state.” That level of assurance is what enables cooperation and collaboration. Developers are able to get infrastructure quickly and Operations is able to provide support because they can reliably ensure their baselines are met.

Enabling Developer Services

At SAP, every development team sets up their own continuous integration and continuous delivery (CI/CD) pipelines. But they rely on the tools and automation that Ran’s group provides. They might use UCP to set up their pipelines or they might access the cookbooks themselves from the group’s internal GitHub repositories and use other tools, such as Jenkins or Travis CI. Regardless of the individual team’s needs, with Chef the CoE is able to provide a flexible solution allowing developers to compose the solution that is right for them.

Although developers cannot make changes to Chef cookbooks directly, they can issue pull requests (PRs), which Martin's team review. Martin says, "Of course, we have standards that the PR must meet and we also have to keep the whole picture in mind. Cookbooks are used by many teams. We can't merge a change that's a special case and that might break something for 20 other teams.

> **"In order to meet the needs of a variety of teams it's essential for organizations to encourage this type of collaboration through code."**
>
> **Martin Schygulla,** DevOps Automation Manager

If it's a pull request that makes sense, then we accept it. If it's not, we need to communicate with the developer about why we rejected it and maybe see if we can implement it in a different way. We welcome contributions to our cookbooks. Everything is open, everyone can see the cookbooks, review them, and create issues or pull requests on that repository." In order to meet the needs of a variety of teams it's essential for organizations to encourage this type of collaboration through code. It's also important to understand where to hand off responsibility.

Martin says, "Developers can, of course, fork a cookbook, rename it, change the owner and then make whatever changes they want, so long as they're willing to be responsible for it."

From back office to the boardroom, warehouse to the storefront, desktop to a mobile device, SAP software empowers people and organizations to work together more efficiently and use business insight more effectively to stay ahead of the competition. SAP applications and services enable more than 345,000 business and public sector customers to operate profitably, adapt continuously, and grow sustainably. SAP knows what it means to enable developers to deliver quickly. We thank the team for taking the time to share their story.