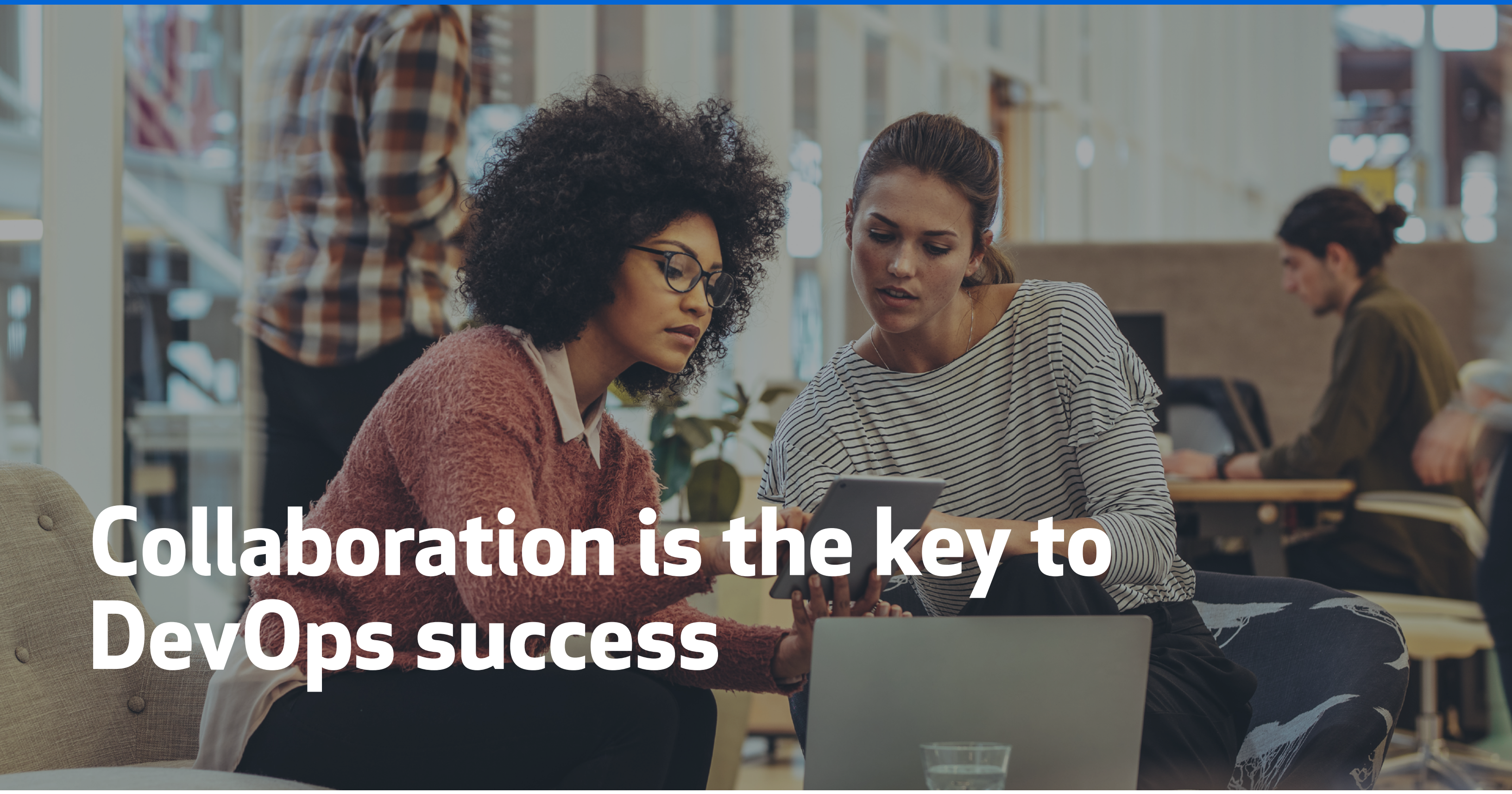
TechTarget | Custom Media
White Paper

GitHub



Collaboration is the key to DevOps success

As one of the world's largest business software companies, SAP has more than 20,000 developers on teams scattered across the globe. In today's DevOps environment, SAP leadership knows that the success of this enormous operation hinges on easy collaboration among far-flung teams and individual developers. That has meant finding an effective way for everyone to communicate and work together, while at the same time driving greater efficiency and creativity.

Dominik Tornow, SAP's director of engineering with the Labs, Processes, and Tools group, believes that centralized processes don't lead to innovation. Developers, architects, and product managers all want to deliver exceptional and continuous functionality. To do this, they need to be able to collaborate on their own terms.

The question is, "how?"

Because DevOps has been so widely accepted, companies and organizations are looking for ways to optimize efficiency from this software development methodology. A central aim is to increase efficiency across the entire systems development lifecycle. And, as research shows, DevOps continues growing rapidly.

DevOps efficiency is about people collaborating, not technology

In a recent global study of IT leaders across 10 major industries, TechTarget reported a 17% year-over-year growth in organizations that have adopted DevOps, leaving only 30% with no DevOps initiatives as of yet. Respondents ranked "eliminating process bottlenecks to speed app release" as the top DevOps challenge, followed closely by "streamlining collaboration among cross-functional DevOps teams." While other challenges further down the list relate to things like software infrastructure, cloud integration, and tools selection, the top two challenges relate to people, not technology.[1]

[1] "Informing the path to enterprise DevOps deployments," TechTarget, October 2018.

In the 2018 version of another major annual study, the percentage of respondents working in a "department called DevOps" grew by a brisk 20%, with the biggest leap occurring in the last year. This global study additionally found that 36% of respondents are currently developing from one-quarter to 100% of their applications by DevOps processes.[2]

Alanna Brown, who launched the annual study above in 2012, doubled down on the "people not technology" challenge of boosting DevOps efficiency. She wrote in a recent blog, "The most important themes for us have always been empowering teams to do their best work, overcoming the cultural divide between development and operations teams and making IT better for everyone."[3]

> In fact, the 2018 report itself notes that best practices in DevOps start "by improving collaboration." The report explains further that, "Cross-team sharing is key to scaling DevOps success. The practices with the most significant impact across the entire DevOps evolutionary journey are dependent on sharing. Organizations that have small pockets of DevOps success, yet never manage to spread that success further, are stalled and cannot progress to higher levels of automation and self-service."[4]

Open source, DevOps, and collaboration

As DevOps adoption has grown in recent years, open source has grown right along with it. One projection of the tremendous interest in open source globally shows revenues doubling in just three years to $26 billion.[5] Collaboration is intertwined with open source, whereby software is often developed in a highly collaborative public process. In fact, global organizations across every industry have been rapidly incorporating open source best practices into how they build and deploy software.

Many organizations are embracing "innersource"—essentially open source behind a firewall. Innersource is a highly collaborative development methodology using best practices from large-scale open source projects. Such projects, like Kubernetes or Microsoft's Visual Studio Code, require coordination across literally thousands of developers and other contributors. Experiences with innersource have resulted in a list of best practices for uniting innersource, open source, and DevOps to drive collaboration and efficiency.

Innersource addresses DevOps challenges by focusing not only on the collaboration of development and operations, but also on the collaboration of all teams in the software process.

"Once you embrace [innersource] and see how new teams come on, you show examples of places where people can contribute and unlock bottlenecks," says Jeremy King, Executive Vice President and Chief Technology Officer for global e-commerce at Walmart. And Jeff Jagoda, Senior Software Engineer at IBM, says that, "We see innersource as a way to improve efficiency through code reuse. But even beyond that, it's an amazing conduit for learning and exchanging ideas and facilitating innovation within IBM."

Team efforts to boost collaboration in DevOps

- **SAP taps into the "wisdom of the crowds"**

Some user organizations are frustratingly jogging in place when it comes to efforts to overcome DevOps challenges. At SAP, with a single group within its huge internal development complex approached management about leveraging an innovative development platform, GitHub, to optimize collaboration among team members spread around the globe.

Initial successes with GitHub spread by word of mouth. Soon, developers on other projects were sharing code and ideas for creating practical workflows, according to SAP's

[2] "2018 State of DevOps Report," Puppet, 2018.

[3] "2018 State of DevOps Report: Practical guidance for your DevOps evolution," Puppet Blog, September 2018.

[4] Ibid(2).

[5] "Projected revenue of open source services from 2017 to 2022 (in billions U.S. dollars)," Statista, 2018.

Tornow. As he reported, the collaborative nature of the platform's "commits, comments, and issues" became core ingredients for efficiency and innovation. In one case, Tornow's group initialized a project containing just a few files on the platform. Within 24 hours, a team 12 time zones away contacted them, asking if they could contribute.

- **C.H. Robinson replaces siloes with collaboration–and superior software**

Facing rapid growth, third-party logistics leader C.H. Robinson needed to ensure its 800-person IT and developer staff worked as a team, despite being spread across numerous offices on three continents. This meant making the team's workflows as collaborative and transparent as possible.

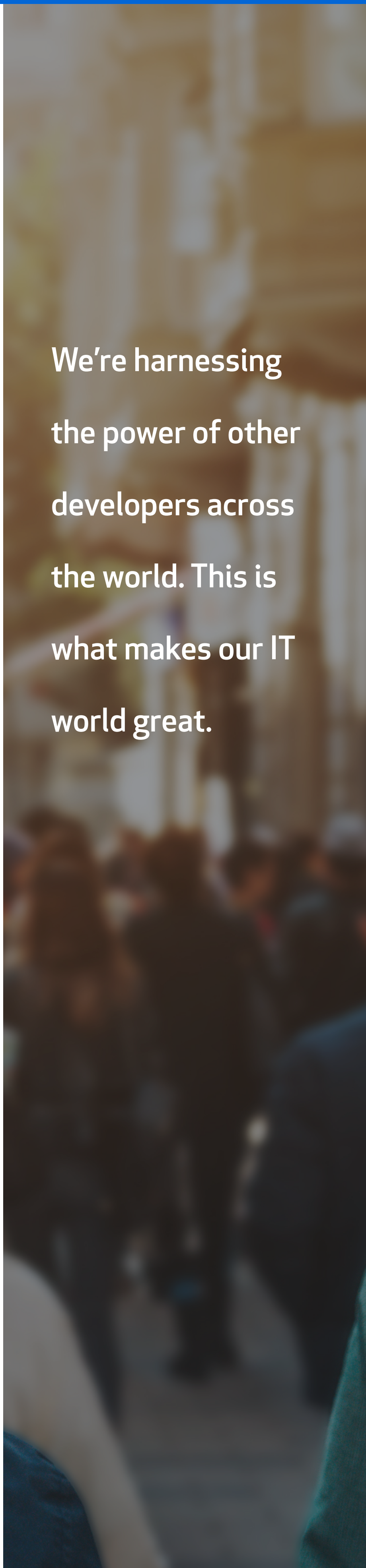
The relative autonomy given to developer teams also gave them freedom to choose their own tools, which caused problems in standardizing code and providing project visibility and documentation. Developers had no good way of working on someone else's code, even if they could locate it. In an effort to empower local development teams, the company had inadvertently created a divided IT organization.

To deal with this situation, C.H. Robinson turned to GitHub. The result to date has been an organization-wide change "towards openness that surfaces the best solutions," according to Tyler Patterson, engineering craft manager. "We're actually doing things internally to advertise code. [The tools] let us be an open and collaborative department and team, and deliver on better products. [...] We're harnessing the power of other developers across the world. This is what makes our IT world great."

- **University of Minnesota taps the power of collaboration from 230 departments and administrative units**

The right collaboration platform also proved to be the engine of efficiency and innovation at the University of Minnesota, where developers are spread across the state in 230 different development departments. As with SAP, a small developer group saw a need for what it called "a social network for collaborating on code." After this group adopted GitHub, the same platform was quickly made available across the entire 60,000-student university.

According to Peter Walz, a system engineer at the university, the platform has resulted in expedited code pushes and more pull requests, the use of which has resulted in significant workflow improvements. "You can easily see what changes are about to made before they are merged in," he says. Walz also reported a reduced impact on IT administration as the platform allowed users to set permissions and self-service on support. Overall, Walz said, GitHub has "simplified cross-departmental collaboration."



We're harnessing the power of other developers across the world. This is what makes our IT world great.

GitHub is the secret sauce for collaboration

As shown, the one thing these three organizations had in common when it came to a selecting a platform to drive collaboration and efficiency to new heights is GitHub. SAP, C.H. Robinson, and the University of Minnesota join millions of organizations and developers worldwide currently using GitHub. Together, these organizations and developers make up the world's largest open source community—sharing code, working securely together, and building better software, faster. Beyond being home to largest open source community, GitHub is also easy to use. As stated by Amir Jaballah, global head of continuous delivery platform at banking giant Societe Generale, "I've rarely seen such enthusiasm for the deployment and adoption of a new solution."

As a highly versatile platform, GitHub can be deployed as on-premises, SaaS, or a hybrid of both with GitHub Connect. Its advanced auditing and monitoring tools let development teams work together and collaborate while meeting critical requirements within the confines of the secure, internal environment.

Whether it is used on-premises with existing servers or in a private cloud, GitHub can help DevOps teams boost overall development efficiency with flexible deployment options, centralized permissions, hundreds of integrations, and technical support.

Key elements of GitHub include:

Collaboration

With pull requests and issues, collaboration and code review are built into the development process. Internal and external teams can share work, discuss changes, and get feedback all in one place. This helps organizations share expertise internally and avoid reinventing field-tested solutions developed for other projects.

Security

Powered by machine learning, security alerts send notifications when vulnerable dependencies are used in a project. Since the launch of security alerts in 2017, over five million alerts have been sent to help the GitHub community keep their projects safe. Token scanning detects and invalidates secrets pushed to GitHub. This enhanced security allows companies to ship when they want to, creating an environment where companies can test and experiment refactored legacy apps and optimize outcomes. Branch protections also ensure changes are reviewed and have gone through CI, GPG verifications, and audit logs.

Integration

GitHub supports custom tools and hundreds of third party apps and services. Using services like Jenkins, CircleCI, Travis CI, or BuildKite, GitHub helps automate workflows according to a team's production environment. The growing number of tools and integrations are accessible at the GitHub Marketplace. They include Sentry for real-time, cross-platform crash reporting and error logging; as well as issue.sh for agile project management inside GitHub, without access to code.

Community

As the largest open source community in the world, GitHub brings the power of open source to teams at work. Teams can tap into new projects, find solutions, and build better software with support from developers around the world.

GitHub is how people build software. Trusted by millions of developers and half of the Fortune 500, GitHub helps DevOps teams of every size collaborate securely—and deliver better customer experiences, faster. To start your free trial or learn more about GitHub's solutions, visit https://github.com/enterprise.