

DevOps erfolgreich umsetzen: Kooperation als entscheidender Faktor

Mehr als 30.000 Entwickler arbeiten in geographisch verteilten Teams für SAP, dem weltweiten Marktführer für Unternehmenssoftware. Das SAP-Management weiß, dass im heutigen DevOps-Umfeld der [Unternehmenserfolg](#) unter anderem von der reibungslosen Zusammenarbeit der Teams und einzelner Entwickler abhängt. Um dies zu ermöglichen, sind effektive Wege der Kommunikation und der Zusammenarbeit für alle Beteiligten erforderlich.

Dominik Tornow, Director of Engineering bei der Gruppe Labs Processes and Tools bei SAP ist der Ansicht, dass zu zentralisierte Prozesse die Innovationskraft eines Unternehmens schwächen. Damit Entwickler, Software-Architekten oder Produktmanager ihre volle Leistung erbringen können, muss man ihnen die Möglichkeit bieten, unter eigenen Bedingungen zusammenzuarbeiten.

Eine DevOps Kultur schaffen:

DevOps ist eine weit verbreitete Methodik im Bereich Softwareentwicklung- und betrieb, deren zentrales Ziel es ist, die Effizienz über alle Phasen der Softwareentwicklung zu steigern.

Eine kürzlich von TechTarget durchgeführte globale Befragung von IT-Führungskräften verzeichnet, bei den Unternehmen die DevOps einsetzen, einen Zuwachs von 17% im Vergleich zum Vorjahr. Nur noch 30 Prozent der Unternehmen führen keine DevOps-Initiativen durch. Die Befragten bezeichneten die „Beseitigung von Prozessengpässen zur Beschleunigung von Releases“ als die wichtigste Herausforderung von DevOps, dicht gefolgt von „Optimierung der Zusammenarbeit zwischen funktionsübergreifenden DevOps-Teams“. Während sich weitere Herausforderungen mit geringerer Priorität auf Aspekte wie Software-Infrastruktur, Cloud-Integration und

Tool-Auswahl beziehen, betreffen die beiden größten Herausforderungen die Kultur, nicht die Technologie.

Alanna Brown, die die jährliche Studie in 2012 lancierte, betont die „Kultur vs. Technologie“-Herausforderung. Sie schrieb in einem kürzlich erschienenen Blog: „Die wichtigsten Themen sind nach wie vor, Teams zu befähigen bestmögliche Arbeit zu leisten und die kulturelle Kluft zwischen Entwicklungs- und Operationsteams zu überwinden.“

Die in 2018 erschienene Studie zeigt, dass Best Practices bei DevOps mit der „Verbesserung der Kollaboration“ beginnt. Die Anwendungsfälle mit den signifikantesten Auswirkungen innerhalb der gesamten DevOps-Entwicklungen beruhen auf teamübergreifenden Austausch von Entwicklungsartefakten.“

Open Source, Inner Source und DevOps

Mit der zunehmenden Verbreitung von DevOps ist auch die Akzeptanz von Open Source gestiegen. [Eine Prognose](#) zum weltweit starken Interesse an Open Source geht davon aus, dass sich der Umsatz in diesem Bereich in nur drei Jahren auf 26 Milliarden Dollar verdoppeln wird. Global agierende Unternehmen aller Branchen haben Open Source-Best Practices in die Entwicklung und das Deployment von Software integriert.

Viele Unternehmen setzen auf „Inner Source“ – was im Wesentlichen die Ausübung von Open Source-Methoden für kommerzielle Softwareentwicklung ist. Open Source-Best Practices und eine Open Source-ähnliche Kultur werden dabei für die Softwareentwicklung innerhalb eines Unternehmens eingeführt. Open Source-Projekte, wie Kubernetes oder Microsofts Visual Studio Code, erfordern die Koordination von buchstäblich tausenden von Entwicklern und anderen Mitwirkenden. Aus diesen Erfahrungen mit Open Source konnten **Best Practices** zur Zusammenführung von Inner Source und DevOps abgeleitet werden.

„Wenn man Inner Source einführt, erlebt man nicht nur wie sich neue Teams bilden und man Menschen die Möglichkeit gibt mitzuwirken, man beseitigt zugleich auch Engpässe“, sagt Jeremy King, Executive Vice President und Chief Technology Officer für Global eCommerce bei Walmart. Und Jeff Jagoda, Senior Software Engineer bei IBM, sagt: „Wir sehen Inner Source als Möglichkeit zur Effizienzsteigerung, bedingt durch die Wiederverwendung von Code. Aber auch darüber hinaus ist es eine hervorragende Möglichkeit Neues zu lernen, Ideen auszutauschen und Innovationen innerhalb von IBM zu fördern.“

Wie eine verbesserte Kollaboration Unternehmen zum Erfolg verhilft

SAP nutzt „Schwarmintelligenz“

Bei der **SAP**, wandte sich eine Gruppe der Entwickler mit dem Vorschlag an das Management, die innovative Entwicklungsplattform **GitHub** zu nutzen um die Zusammenarbeit zwischen den geographisch verteilten Teams zu optimieren. Rasch konnten erste Erfolge, bedingt durch die Einführung von GitHub, verzeichnet werden. Dies führte dazu, dass auch Entwickler aus anderen Projekten ihren Code und ihre Ideen für die Erstellung agiler Workflows teilten, wie Dominik Tornow, Director of Engineering bei SAP in der Gruppe Labs Processes and Tools, berichtet. Der kollaborative Charakter der „Commits, Comments und Issues“ von GitHub wurde zu einem wichtigen Element um Effizienz und Innovationskraft im Unternehmen zu steigern.

C.H. Robinson bricht mit dem Silodenken und setzt stattdessen auf Kollaboration – und bessere Software

C.H. Robinson zählt zu den führenden Logistik-Drittanbieter weltweit. Im Zuge einer rapiden Wachstumsphase legte das Management besonderen Wert darauf, den 800 Developern und IT-Mitarbeitern trotz der Expansion zu ermöglichen optimal zusammenzuarbeiten. Da lokale Entwicklerteams unterschiedliche Tools verwendeten, kam es zu Problemen bei der Standardisierung von Code und der Transparenz und Dokumentation von Projekten. Letzten Endes hatte dies zu einer fragmentierten IT-Organisation geführt. Um die Silos aufzubrechen, setzte C.H. Robinson auf GitHub und ein agiles Projektmanagement-Tool. Das Ergebnis ist ein unternehmensweiter Wandel „in Richtung Offenheit, die die besten Lösungen hervorbringt“, so Tyler Patterson, Engineering Craft Manager. „Wir unternehmen intern einiges, um Code bekannt zu machen. Die Tools ermöglichen uns ein offenes Vorgehen, indem wir die Barrieren zwischen IT, Code, Versionsverwaltung niederreißen.“

GitHub ist der Schlüssel zu einer erfolgreichen Kollaboration

Sowohl SAP als auch C.H. Robinson haben sich für **GitHub** entschieden um die Effizienz und Kollaboration zu fördern. Damit gehören beide Unternehmen zu den 2,1 Millionen Organisationen und 31 Millionen Entwicklern weltweit, die derzeit GitHub nutzen. Amir Jaballah, Global Head of Continuous Delivery Platform **beim Bankenriesen Société Générale**, erklärt: „Ich habe selten so viel Begeisterung für die Implementierung und Einführung einer neuen Lösung erlebt.“

GitHub kann on-prem, als SaaS oder als Hybridlösung (GitHub Connect) eingesetzt werden. Die Auditing- und Monitoring-Tools ermöglichen den Einsatz in hoch regulierten Industrien mit hohen Compliance-Anforderungen.

Die GitHub-Plattform adressiert folgende DevOps-Elemente:



Zusammenarbeit

Bei Pull-Requests und Issues sind Kollaboration und Code-Review in den Software-Entwicklungsprozess integriert. Interne und externe Teams können Arbeit teilen, Änderungen besprechen und Feedback erhalten – alles an einem Ort. Dies hilft Unternehmen, Fachwissen intern auszutauschen und zu vermeiden, dass praxiserprobte Lösungen, die für andere Projekte entwickelt wurden, neu erfunden werden.



Community

2,1 Millionen Organisationen und über 40 Millionen Entwickler vertrauen auf GitHub. Teams können an neuen Projekten mitwirken, Lösungen finden und mit der Unterstützung von Entwicklern auf der ganzen Welt bessere Software entwickeln.



Sicherheit

Durch KI-gestützte Prozesse werden Sicherheitslücken identifiziert, die in automatischen Sicherheitswarnungen resultieren, wenn gefährdete Abhängigkeiten in einem Projekt genutzt werden. Seit der Einführung von Sicherheitswarnungen im Oktober 2017 wurden über fünf Millionen Warnungen verschickt, um die GitHub-Community dabei zu unterstützen, die Sicherheit ihrer Projekte zu gewährleisten. Das Token-Scanning-Feature von GitHub erkennt und invalidiert Zugangsdaten, die an GitHub übertragen wurden. Maßnahmen zum Schutz von Git-Branched stellen sicher, dass alle Code-Änderungen durch Testfälle überprüft werden und Git-Commits kryptographisch signiert wurden.



Integration

GitHub unterstützt kundenspezifische Werkzeuge und hunderte von Drittanbieteranwendungen und -diensten. Mit Diensten wie Jenkins, CircleCI, Travis CI oder Azure DevOps unterstützt GitHub die Automatisierung von Workflows entsprechend ihrer Produktionsumgebung. Die wachsende Anzahl von Tools und Integrationen ist über den [GitHub Marketplace](#) zugänglich. Dazu gehören Sentry für plattformübergreifendes Crash-Reporting in Echtzeit und Fehlerprotokollierung sowie issue.sh für agiles Projektmanagement innerhalb von GitHub ohne Code-Zugriff.