



## Software-Produktion im Team

# Wenn es mal wieder schnell gehen soll: DevOps

Neue Software und IT-Services müssen heute sehr schnell an den Markt kommen.

Entwicklungsabteilungen und Betrieb stehen daher unter hohem Zeitdruck. Von Jiří Petr

Die Geschwindigkeit, mit der Unternehmen neue Angebote bereitstellen können, ist zu einem der wichtigsten Schlüsselfaktoren für den Geschäftserfolg geworden. Wer nicht imstande ist, gewinnbringende Ideen sehr schnell umzusetzen, wird von der Konkurrenz links überholt. Damit das nicht passiert, müssen interne Entwicklungsabteilungen bzw. externe IT-Dienstleister in der Lage sein, die Herstellung und den Betrieb von Software-Lösungen viel schneller als früher zu realisieren. In der Praxis ist es inzwischen so, dass Services, für die früher etwa ein halbes Jahr Zeit blieb, in ein oder zwei Monaten zur Verfügung stehen müssen. Das bedeutet, dass es zwischen Entwicklung und Betrieb an keiner Stelle zu Spannungen kommen darf – von der Aufnahme der Anforderungen über die Entwicklung und das Testen bis zur Auslieferung.

Grössere KMU und Grossunternehmen verfügen aber in der Regel über eine heterogene Entwicklungsumgebung. Neben Java- und .NET-Plattformen werden nicht selten Webtechno-

logien und proprietäre Entwicklungssysteme parallel eingesetzt. Ein geeignetes Mittel, um die von Fachabteilungen und Vertrieb verlangte schnelle Time-to-Market einhalten zu können, ist die Überführung in den DevOps-Modus. Um die enge Zusammenarbeit von Entwicklung (Development) und IT-Betrieb (Operations) in der Praxis wirksam zu nutzen, ist zwingend ein höherer Reifegrad in Sachen Automation nötig.

Wie wird nun «DevOps» im Unternehmen umgesetzt und wo liegen die Knackpunkte? Eins ist sicher: Damit man neue Services schneller mit Software abbilden und an den Markt bringen kann, reicht der Einsatz von «Tools und Toys» nicht aus. Im Gegenteil: Ein Wildwuchs von DevOps-Systemen kann die Koordination der Aufgaben und die Verbesserung der Prozessabläufe hemmen. Bei grösseren Firmen setzen aber oftmals unterschiedliche Entwicklergruppen verschiedene DevOps-Tools ein. Hinzu kommt, dass die Werkzeuge, ob Open Source oder kommerziell, zwar immer mächtiger

werden, ein umfassendes Werkzeug zumindest zum heutigen Zeitpunkt aber noch gar nicht erhältlich ist. Tool-Hersteller versuchen schon, zunehmend mehr Funktionalitäten in ihre Produkte zu packen. Den DevOps-Prozess mit einer einzigen Lösung abzubilden, ist indes bis dato noch nicht möglich.

## Konzept mit Zieldefinition

Umso wichtiger ist es für Entwicklungsleiter, auf Unternehmensebene ein Konzept zu erarbeiten und die unterschiedlichen Ansätze zu vereinheitlichen. In diesem Konzept ist klar zu definieren, wie Continuous Server (zum Beispiel Jenkins) die kontinuierliche Integration (Continuous Integration) und Auslieferung (Continuous Delivery) neuer Software-Bestandteile erlauben und welches Tool-Set insgesamt unternehmensweit von allen Zulieferern aus den unterschiedlichen Entwicklungsteams eingesetzt wird.

Dabei ist es wichtig, auch Ausnahmen von der Regel zuzulassen. Insbesondere gilt es bei Legacy-Systemen zu beachten, wie diese in den Ablauf eingebunden werden können, gestaltet sich doch hier die Bereitstellung von Testdaten und die Automatisierung der Tests meistens schwierig. Bei aller Vereinheitlichung ist also mit Augenmass vorzugehen und es sollten möglichst viele Einheiten integriert werden. Wenn viele Umgebungen zu verwalten sind, ist der Aufwand entsprechend höher.

Stehen einzelne Teams oder Entwickler der Veränderung kritisch gegenüber, hilft es, klar den Nutzen aufzuzeigen. Wichtig ist vor allem, das Team so früh wie möglich in die Konzeptionsphase einzubinden und dabei deren eigene Ideen ernst zu nehmen. Konzepte, in denen sich Mitarbeitende in irgendeiner Form wiederfinden, werden eher mitgetragen und gelebt. Insgesamt empfiehlt es sich, je nach Situation, eine Roadmap zu definieren, die aufzeigt, wie und vor allem wie weit ein DevOps-Konzept umgesetzt werden soll und welche Ziele das Team damit erreichen will.

## Kollaborative Kultur

Neben der Tool-Auswahl kommt es auch darauf an, neue Funktionseinheiten, die man in Betrieb nehmen will, zügig zu spezifizieren und diese so klein wie möglich zu halten. Um die geforderte Geschwindigkeit bei der Auslieferung zu erreichen, müssen die Tests viel früher als bislang durchgeführt und vor allem klar definiert sowie automatisiert werden. Das Ziel von DevOps besteht ja gerade darin, mit möglichst vollautomatisierten Abläufen und im Hintergrund durchgeführten Tests die Qualität der Software zu verbessern, die Durchlaufzeit zu optimieren und einzelne Software-Blöcke automa-

### DevOps-Konferenz im November

**DevOpsCon:** Die Konferenz für Continuous Delivery, Micro-Services, Docker, Clouds & Lean Business informiert über innovative Infrastruktur und moderne Unternehmenskulturen, deren Zusammenspiel zukunftsfähiges Lean Business ermöglicht.

**München: 20.-23. November, Expo: 21.-22. November 2017**  
Infos: [www.DevOpsconference.de](http://www.DevOpsconference.de)



## «Ein Wildwuchs von DevOps-Systemen kann die Prozessabläufe hemmen»

Jiří Petr

tisiert bereitzustellen. Leider wird nach wie vor auch in grösseren Unternehmen noch sehr viel manuell getestet. Das birgt die Gefahr, dass sich die Qualität eher verschlechtert als verbessert.

DevOps beschränkt sich aber nicht darauf, Werkzeuge zur Automatisierung zu definieren und die Mitarbeiter zum Gebrauch derselben zu befähigen. Vielmehr muss die Idee in den Köpfen der Entwickler und auch im Betrieb verankert werden. Das Ziel lautet: Durch verbesserte Abläufe und abteilungsübergreifende Zusammenarbeit qualitativ höhere Produkte bedeutend schneller auszuliefern. Voraussetzung dafür sind zuallererst eine kollaborative Unternehmenskultur und gemeinsame Ziele.

## Tools, Automation und Budget

Erst danach kommt die Auswahl der richtigen Tools, die Testautomation und die Einbindung von Legacy-Systemen. Ein DevOp-Plan mit Milestones, Ownern und klar definierten Lieferzielen ist ebenso unerlässlich wie Management, Standardisieren und Automatisieren komplexer Umgebungen sowie der Aufbau des dazu nötigen Skillsets. Darüber hinaus gewinnen unter dem Oberbegriff «DevSecOps» die Sicherheitsfragen der IT-Infrastruktur an Bedeutung.

Allein die Bereitschaft, eine solche neue Kultur im Unternehmen zwischen Fachabteilungen, Entwicklern und Betrieb zu etablieren, reicht indes nicht aus. Denn der Einsatz beispielsweise von Open-Source-Werkzeugen bedeutet noch lange nicht, dass keine Kosten anfallen. Es braucht für eine erfolgreiche Umsetzung der DevOps-Philosophie ein Budget und insbesondere die Unterstützung auf Executive-Level zur Bereitstellung finanzieller und personeller Ressourcen. ■

Jiří Petr ist CEO des Software-Entwicklungsunternehmens Intersys: [www.intersys.ch](http://www.intersys.ch)