

# DevOps

## Drum teste, wer in SAP ausliefert

Häufiger, aber kleiner – dank DevOps sinkt das Risiko pro Release, die verfügbare Zeit für die Qualitätssicherung jedoch ebenso. DevOps in SAP-Umgebungen mit ihren Abhängigkeiten muss um automatisierte Testverfahren ergänzt werden.

Von Achim Töper, Basis Technologies



**Achim Töper,**  
Senior DevOps  
Solution Specialist,  
Basis Technologies.

DevOps (Development und IT Operations) gewinnt in der Softwareentwicklung zunehmend an Bedeutung – auch im SAP-Umfeld. DevOps ermöglicht es Unternehmen, auf die sich schneller ändernden Marktanforderungen agil zu reagieren.

Bitte beachten Sie auch den Community-Info-Eintrag Seite 71



**M**an könnte es das „Risiko-Paradoxon“ bei DevOps nennen: Selbst die komplexesten Entwicklungsprojekte erfolgen in vielen kleinen Schritten. Dadurch sinkt das Risiko pro Release, da weniger Code-Änderungen logischerweise weniger Fehler mit sich bringen. Andererseits sind die einzelnen Releasezyklen so kurz, dass insgesamt weniger Zeit für die Qualitätssicherung bleibt. In der Regel dauert ein Sprint – also einer von vielen iterativen Arbeitsgängen zur Entwicklung neuer Funktionalität – inklusive Tests und Deployment auf den Produktivsystemen nur zwei Wochen. Demgegenüber folgt im klassischen Wasserfall-Modell der Entwicklungsarbeit eine ausgedehnte Testphase, erst dann wird das Release freigegeben.

SAP-Entwickler und -Administratoren haben ein grundlegendes Ziel: Sie müssen sicherstellen, dass das, was gestern funktioniert hat, auch morgen noch funktionieren wird. Was insbesondere in komplexen Umgebungen ebenso erfolgskritisch wie schwierig ist. Denn hier besteht eine schier endlose Zahl von Abhängigkeiten, sodass schon kleine Fehler verheerende Folgen haben können.

Damit DevOps in SAP-Umgebungen ein Erfolg wird, wird nicht nur eine integrierte Tool-Chain für Continuous Delivery (CD) und Continuous Integration (CI) benötigt. Vielmehr sind zusätzlich Tools für Continuous Testing nötig. Damit einher geht eine neue Teststrategie in fünf Schritten:

Erstens muss das Shift-Left-Prinzip auch auf die Qualität angewandt werden. Fehler zu beheben, noch bevor ein neues Release in Betrieb geht, sorgt nicht nur für weniger Fehlfunktionen und Serviceunterbrechungen. Vielmehr werden damit Kostenreduktionen bis um den Faktor 15 möglich. Zu diesem Zweck müssten Tests aber schon in früheren Projektphasen als bisher stattfinden.

Das hat zweitens unmittelbare Auswirkungen auf den Entwicklungsprozess selbst. Der neue Code muss während eines Sprints mehrfach getestet werden, und zwar nicht nur hinsichtlich der Voll-

ständigkeit und der Funktionalität, sondern auch hinsichtlich seines Verhaltens in den Produktivumgebungen. Peer-Reviews, Retrospektiven und Messungen gehören ebenfalls zu einer kontinuierlichen Qualitätssicherung dazu und erlauben eine kontinuierliche Verbesserung der Testverfahren.

Das bedeutet drittens, dass die Unternehmen die Qualitätssicherung in ihre funktionsübergreifenden DevOps-Teams und -Prozesse einbetten sollten. Nicht nur die für den Betrieb Verantwortlichen, sondern auch die Tester müssen Teil des DevOps-Teams sein und an allen Schritten eines Sprints mitwirken.

Wegen der Komplexität von SAP-Umgebungen droht die Qualitätssicherung zu einem Flaschenhals zu werden. Deshalb brauchen viertens DevOps-Teams die Unterstützung durch Tools für automatisierte Regressionstests. Geeignete Tools decken dabei praktisch die gesamte Produktivumgebung ab und liefern daher Ergebnisse, die so realitätsnah sind, dass positiv getesteter Code mit einem stark reduzierten Risiko implementiert werden kann.

Fünftens brauchen SAP-Bestandskunden eine flexible Deployment-Strategie. Die Projektverantwortlichen müssen abhängig von den Testergebnissen dynamisch entscheiden können, welche Änderungen am Ende eines Sprints in die Produktivumgebung übernommen werden sollen und welche nicht. Auch hierfür benötigen sie die Unterstützung von Werkzeugen zur Prozessautomatisierung.

Eines der zentralen Versprechen des DevOps-Konzepts lautet höhere Code-Qualität. Kürzere Releasezyklen reichen jedoch allein nicht aus, um das Risiko von Code-Fehlern in Produktivumgebungen einzudämmen. Gerade SAP-Teams müssen deshalb das Testen zum integralen Bestandteil ihrer Prozesse machen, um DevOps auch in komplexen Umgebungen zum Erfolg zu führen. Das gelingt jedoch nur mithilfe einer integrierten Tool-Chain, die hochautomatisierte Testwerkzeuge miteinschließt.