



□ Michael Hüttermann

(michael@huettermann.net)

ist Oracle Java Champion, unterstützt als Delivery Engineer Projekte und gibt Seminare in den Bereichen DevOps, Continuous Delivery und Agile ALM. Seine Bücher „Agile ALM“ (2011) und „DevOps for Developers“ (2012) sind international die ersten, die sich mit den jeweiligen Themen auseinandersetzen. Mehr dazu unter <http://huettermann.net>.

## objektspektrum themenspecial: DevOps

# DevOps – hier rückt zusammen, was zusammengehört: Entwicklung und Betrieb

Schlechte Nachrichten für Early Adopter! Wer sich heute mit den Themen Continuous Delivery und DevOps beschäftigt, gehört nicht mehr zu den Allerersten! Andersherum wird ein Schuh draus. Haben diese Themen auch im Frühling 2014 noch immer nicht Ihre gänzliche Aufmerksamkeit gewonnen? Möchten Sie wirklich nicht von kürzeren Durchlaufzeiten, besserer Kollaboration zwischen Entwicklung und Betrieb sowie erhöhter Mitarbeiter- und Kundenzufriedenheit profitieren? Wenn doch, dann ist dieses Themenspecial genau das Richtige für Sie.

Continuous Delivery beschreibt einen heterogenen Baukasten aus Good Practices, um Ihre Softwareentwicklungs- und Bereitstellungsprozesse ganzheitlich zu optimieren. Ein Teilaspekt davon ist DevOps, ein Kofferwort aus Development (Entwicklung) und Operations (Betrieb), also gemeinsame Ziele, Prozesse und Werkzeuge, die sowohl in der Entwicklung als auch im Betrieb zum Einsatz kommen.

Continuous Delivery hat viele Vorteile. Einige davon sind:

- geringere Entwicklungs- und Bereitstellungskosten,
- mehr Outcome bei verkürzter Cycle-Time,

- weniger reaktive Aktivitäten (z. B. Firefighting),
- dafür mehr proaktive Aktivitäten (z. B. Innovationen vorantreiben) sowie
- eine höhere Mitarbeiter- und Kundenzufriedenheit.

Traumhaft! Doch wie ist all das zu erreichen?

Zunächst einmal ist ein Umdenken gefragt. Schluss mit Mikrooptimierungen! Ein „Continuous Delivery“-Vorgehen ist mit einer ganzheitlichen Herangehensweise verknüpft. Dabei werden holistische, treibende Ziele, sogenannte KPI's (Key Performance Indicators), identifiziert und verfolgt.

Um übergreifend das Ganze zu sehen, kommen verschiedene Ansätze zum Einsatz, zum Beispiel Praktiken aus dem Systems Thinking, Agilen ALM, Theory of Constraints oder Lean. Einmal das Ganze im Blick müssen kontinuierliche Rückkopplungen sowie Zeit für Innovation und Experimente in das Vorgehen eingeflochten werden.

Als Vehikel bei der Umsetzung kommen häufig Value Stream Maps zum Einsatz, die Hand in Hand mit dem Design einer Continuous Delivery Pipeline fortgeschrieben werden. Die Pipeline, eine Metapher für eine Fertigungsstraße, nimmt Changes auf, z. B. neue Kundenanforderungen, und begleitet sie bis zur Verfüg-

barkeit für den Nutzer auf Produktionsmaschinen. Einzelne Stationen sind durch Quality Gates angereichert.

Zahlreiche Praktiken kommen zum Einsatz. Beispielsweise wird von einem hohen Automationsgrad Gebrauch und Konfigurationsmanagement explizit gemacht, z. B. durch „Infrastructure as Code“. Der Fluss kann weiter erhöht werden durch task-basierte Entwicklung und eine minimale Anzahl von Branches im Versionskontrollsystem.

Ein essenzieller Baustein einer „Continuous Delivery“-Initiative ist der Einsatz von DevOps. Auch wenn DevOps für sich genommen bereits alleine zum Einsatz kommen kann, ist es andersherum ausgeschlossen: ohne DevOps kein Continuous Delivery!

Genauso wie Continuous Delivery und DevOps einen bunten Strauß aus den verschiedensten Ansätzen darstellen, diskutiert auch dieses Themenspecial eine

Vielzahl unterschiedlicher Themenbereiche und Ansätze.

**Thomas Rümmler** und **Christian Schlag** beschreiben in ihrem Artikel wie DevOps und Continuous Delivery genutzt werden können, um sich gemeinsam zu verbessern. **Justin Vaughan-Brown** widmet seinen Artikel einem Schwerpunkt von DevOps, nämlich der Service-Virtualisierung, um Anwendungen schnell, kostengünstig und hochwertig entwickeln zu können.

**Michael Brokmann** und **Robert Michel** haben für ihren Beitrag die Cloud als Eckpfeiler einer Continuous Delivery Pipeline gewählt. **Brian Rosenberger** liefert einen Erfahrungsbericht aus der Praxis. Er stellt dar, wie DevOps und Continuous Delivery genutzt werden.

**Sebastian Müller** beschreibt das veränderte Rollenverständnis, das DevOps in der kompletten IT nach sich zieht. Er thematisiert, wie die DevOps-Kultur die Ar-

beitsweise in IT-Abteilungen grundlegend verändert am Beispiel von Autoscout 24.

**Matthias Marschall**, vielen bekannt als ein deutscher DevOps-Vorreiter, gibt Ihnen einen Überblick über das „Value Stream Mapping“ und wie es helfen kann in Richtung Continuous Delivery zu optimieren. Schließlich liefert **Marco Schulz** einen Übersichtsbeitrag zu „Continuous Delivery“ und illustriert die Aspekte, die nach seiner Erfahrung maßgeblich sind.

Und nun wünsche ich Ihnen viel Spaß bei der Lektüre und alles Gute bei Ihrer ganz persönlichen Continuous Delivery- und DevOps-Initiative. ■

Ihr

*Michael Hüttermann*

*Michael Hüttermann*