präsentiert von: heise Developer
www.heise-developer.de

Mehr als drei Buzzwords:
Continuous Delivery, DevOps und Container

iX DEVELOPER

2/2016

Auf der Heft-DVD

Massig Material für Entwickler

Software:
Ubuntu, Eclipse, Docker, Kubernetes, Ansible, Chef, Puppet, Jenkins, Vagrant, Mesos, VirtualBox etc.

Multimedia:
4 Continuous-Lifecycle-Videos zu Continuous Delivery, DevOps und Branch-Modellen; mehrere Episoden des SoftwareArchitek-TOUR-Podcasts

Literatur:
Microservices (Broschüre), Continuous Delivery (Buchauszüge), *iX* Developer „Bessere Software" (Sonderheft)

Effektiver entwickeln

Container:

Docker & Co. revolutionieren die IT-Landschaft

Grundlagen, Ökosystem und Orchestrierung

DevOps:

IT-Kulturwandel eint Entwickler und Administratoren

Compliance, Sicherheit, APM und Change Management

Continuous Delivery:

Basiswissen, Einführung, Tools und Management

Chef, Puppet und Ansible im Dienst der Continuous Integration

Microservices:

Architekturparadigma stützt Continuous Delivery und DevOps

Mannheim, Congress Center Rosengarten,
14.-16. November 2016

continuous lifecycle 2016

Prozesse – Tools – Erfahrungen

CONTINUOUS WHAT? CONTINUOUS EVERYTHING!

Die drei Auflagen der Continuous Lifecycle haben es deutlich gezeigt: Continuous Delivery, DevOps und Containertechniken sind in der Praxis angekommen – nicht nur in Web-2.0- und Cloud-Firmen, sondern auch in Unternehmen, die klassische Unternehmens- oder Embedded-Anwendungen herstellen.

Wie aber setzt man diese Konzepte und die damit verbundenen Techniken sinnvoll im eigenen Projekt oder Unternehmen ein? Wo steckt das größte Potenzial, wo liegen die Fallstricke? Antworten auf diese Fragen gibt Ihnen die Continuous Lifecycle – die wichtigste deutsche Konferenz zu Continuous Delivery, DevOps und Containerisierung. Ganzheitlich widmet sie sich den Konzepten, Prozessen und Werkzeugen hinter Continuous Delivery, DevOps und Co. und bietet Erfahrungen, die Ihnen praktisch weiterhelfen.

Call for Papers bis 30. Mai 2016

THEMEN (Auszug)

- Der richtige Umgang mit Continuous Delivery
- Praktische Umsetzung von DevOps-Methoden
- Werkzeuge für agiles Application Lifecycle Management (Versionskontrolle, Continuous Integration, Ticketing und Bugtracking)
- Containerisierung mit Docker und den Werkzeugen aus dem Docker-Ökosystem
- Build Management
- Code-Reviews
- Testen und Qualitätssicherung
- Betrieb und Monitoring
- Fallstricke und Best Practices verteilt arbeitender Software-Teams

ZIELGRUPPE

- Softwareentwickler
- Softwarearchitekten
- Administratoren
- Projektleiter
- IT-Strategen

Gold-Sponsor:

codecentric XebiaLabs Deliver Faster

Silber-Sponsoren:

adesso business. people. technology. CloudBees The Enterprise Jenkins Company OPITZ CONSULTING

Veranstalter:

iX heise Developer dpunkt.verlag

www.continuouslifecycle.de

Der ewige Kreis

Die Softwareentwicklung zeigt derzeit Mut zur Größe – zur kleinen Größe. Der Trend geht weg von monolithischen Systemen hin zu kleinen Bausteinen wie Microservices. Auch die Zeiträume zwischen Releases schrumpfen: Statt alle ein bis drei Jahre ein großes Major Release herauszubringen und durch Service Packs zu ergänzen, veröffentlichen Unternehmen mithilfe von Continuous Delivery eine neue Softwareversion, sobald sich eine Komponente ändert. Große Internetfirmen wie Google, Amazon oder Facebook aktualisieren ihre Plattformen inzwischen im Minutentakt. Dass die Nutzer davon nur bei Änderungen der Oberfläche etwas mitbekommen, zeigt, wie reibungslos die kontinuierlichen Prozesse funktionieren können. Ein Wartungsfenster am Wochenende existiert im Internet nicht. Die Softwareentwicklung wird zum ewigen Kreis.

Container bieten Entwicklern eine in sich abgeschlossene und damit überschaubare Umgebung zum Testen und Ausrollen der Anwendungen. Dank der einfachen Reproduzierbarkeit können die Teams auf große Testumgebungen verzichten. Ändern sich die Anforderungen oder Umgebungen, passen sie einfach die Schablone an und rollen neue Container aus.

Containerisierung, Komponenten oder kontinuierliche Prozesse sind keine Erfindung der 2010er-Jahre, sondern haben in der ein oder anderen Ausprägung immer wieder den Weg in die Schlagzeilen geschafft. Oft scheiterten sie in der Realität jedoch an Komplexität und Inkompatibilitäten. Es gibt jedoch entscheidende Unterschiede, die den aktuellen Techniken und Tools zum Erfolg verhelfen sollten.

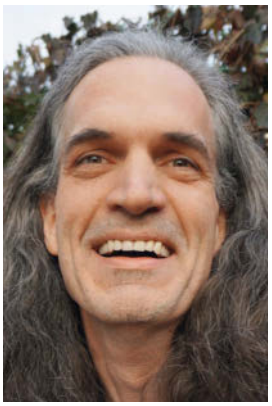
Microservices punkten durch die Unabhängigkeit der einzelnen Dienste voneinander: Jeder hat seine eigene Ablaufumgebung und kommuniziert über Standardnetzwerkprotokolle mit der Außenwelt. Frühere Modularisierungskonzepte bedienten oftmals eine spezielle Sprache oder verwendeten einen Applikationsserver und scheiterten an der Zusammenarbeit über die Grenzen hinweg. Docker-Container haben einen klaren Fokus auf Entwickler und sind leichter zu verwenden als vorherige Ansätze. Jenkins ist besonders wegen seiner Flexibilität und des offenen Plug-in-Konzepts zur festen Größe für Continuous Delivery geworden. Bei den Konfigurations-Management-Systemen buhlen vor allem Chef, Puppet und Ansible mit unterschiedlichen Stärken um die Gunst der Anwender.

Die technische Basis und die weitgehend quelloffenen Werkzeuge sind jedoch nur die halbe Miete. DevOps, also das Zusammenspiel von Softwareentwicklern (Dev) und den Administratoren (Ops), erfordert ein Umdenken der Beteiligten: Die Fraktionen müssen zusammenarbeiten und bei Problemen gemeinsam nach Lösungen suchen. Wenn Entwickler im Prozess der kontinuierlichen Verteilung eine Komponente ändern und damit einen neuen Build anstoßen, der die Änderung sofort ins Livesystem übernimmt, kommt ihnen und ihrem Werk zudem eine höhere Bedeutung zu als bisher. Der Satz „You build it, you run it“ von Amazons CTO Werner Vogels bringt es auf den Punkt: Jeder Entwickler ist für seinen Code verantwortlich und bekommt dadurch mehr Bezug zum Gesamtprodukt.

Der Kulturwandel erfordert Umdenken auf allen Ebenen. In einigen Projekten erweist es sich als sinnvoll, dass sich das Management aus dem Tagesgeschäft heraushält und somit den einzelnen Teams mehr Verantwortung zuteilt. Auch traditionelle Unternehmen wie Maschinenbauer können durchaus von moderner Softwareentwicklung profitieren und mit den passenden Ansätzen den Produktionsbetrieb in der Fabrik optimieren, ohne dort den reibungslosen Ablauf zu gefährden. Damit schließt er sich wieder, der ewige Kreis.

Mit diesem Heft möchten wir Ihnen helfen, sich bei den Themen Continuous Delivery, DevOps, Containerisierung und Microservices zurechtzufinden. *iX* und *heise Developer* wünschen Ihnen viel Spaß beim Lesen!

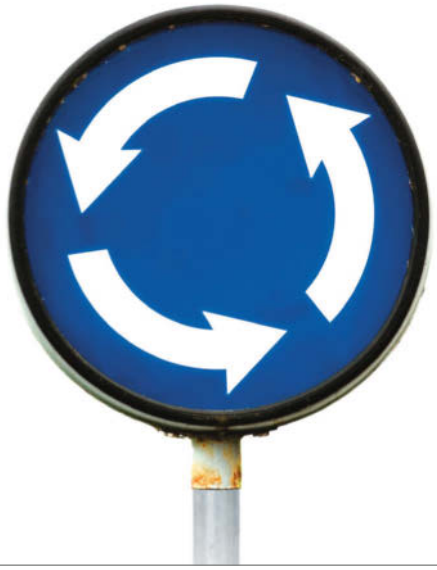
RAINALD MENGE-SONNENTAG & ALEXANDER NEUMANN

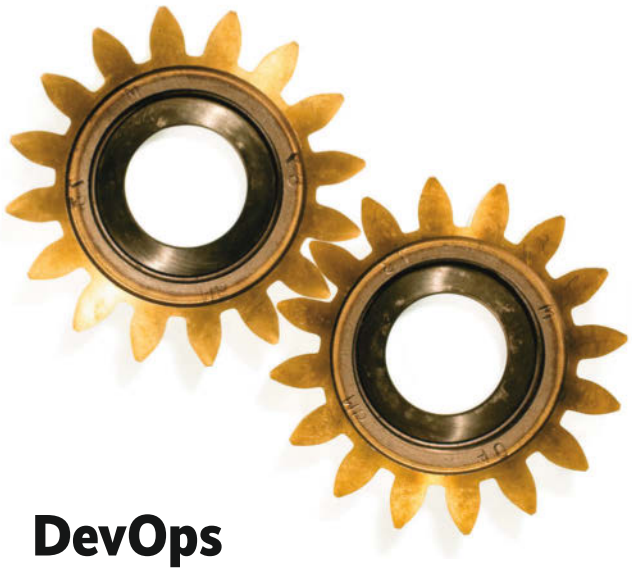


Continuous Delivery

... verspricht Produktivsetzungen der entwickelten Software auf Knopfdruck bei besserer Qualität. Was sind die zentralen Ideen dahinter, welche Voraussetzungen muss man schaffen, um damit erfolgreich zu sein, und was sind die wichtigsten Tools?

ab Seite 36





DevOps

... lebt nicht nur von geeigneten Werkzeugen, sondern vor allem von gegenseitigem Vertrauen und geteilter Verantwortung. Erst dann lässt sich die Idee hinter DevOps umsetzen, wie Erfahrungen mit entsprechenden Tools aus der Praxis zeigen.

ab Seite 66

Einführung

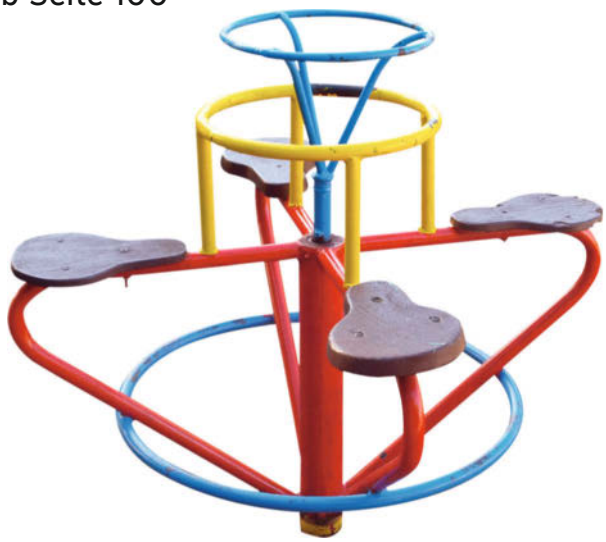
Continuous Delivery

DevOps

Microservices

... lassen sich unabhängig skalieren, und der Ausfall eines Service beeinflusst die anderen nicht. Je kleiner er ist, desto größer der Vorteil. Aber wo liegt die Grenze für die Größe eines Microservice? Und wann ergeben sie überhaupt Sinn?

ab Seite 100



Docker

... vereinfacht den Test- und Produktivbetrieb. Die Docker-eigenen Werkzeuge decken die Grundbedürfnisse ab, darüber hinaus ist inzwischen ein reiches Ökosystem zur Orchestrierung der Container in nahezu beliebig komplexen Szenarien gewachsen. Best Practices sorgen für einen sicheren Betrieb.

ab Seite 126

Microservices

Container

Sonstiges

Alle Links: www.ix.de/ix1614004

Artikel mit Verweisen ins Web enthalten am Ende einen Hinweis darauf, dass diese Webadressen auf dem Server der *iX* abrufbar sind. Dazu gibt man den *iX*-Link in der URL-Zeile des Browsers ein. Dann kann man auch die längsten Links bequem mit einem Klick ansteuern. Alternativ steht oben rechts auf der *iX*-Homepage ein Eingabefeld zur Verfügung.