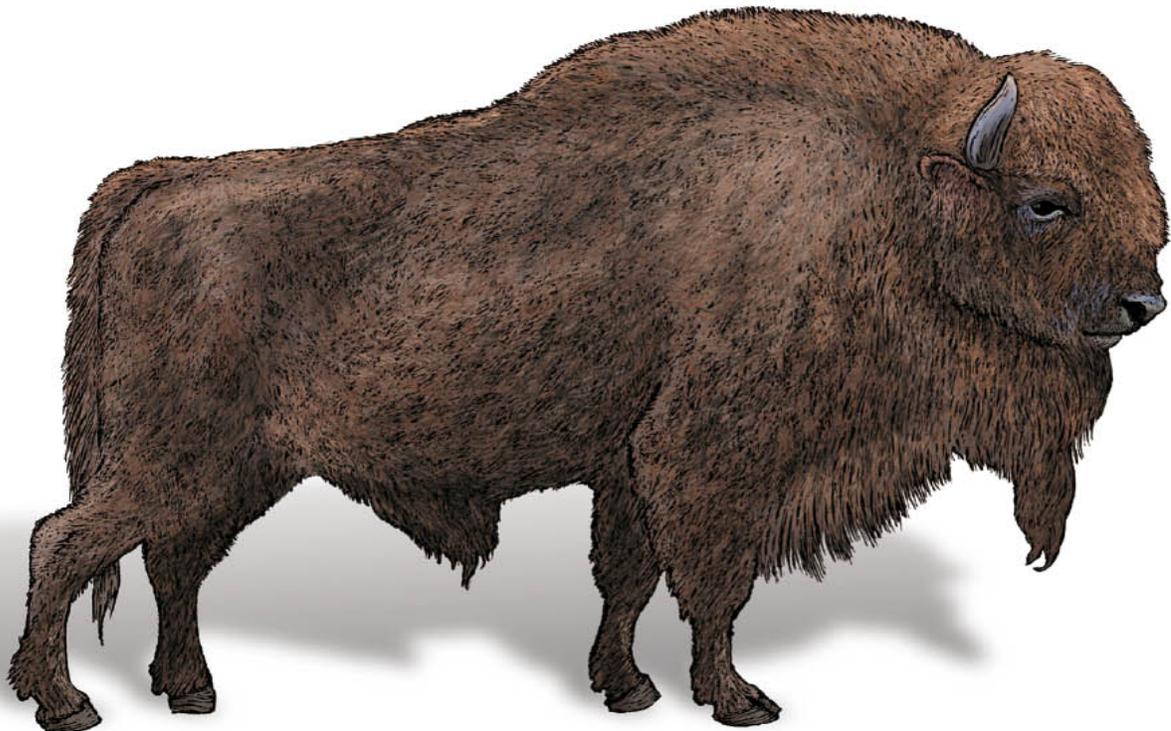


O'REILLY®

2. Auflage  
Aktualisiert  
und erweitert

# Das DevOps Handbuch

Teams, Tools und Infrastrukturen  
erfolgreich umgestalten



Gene Kim, Jez Humble,  
Patrick Debois, John Willis & Nicole Forsgren  
Übersetzung von Thomas Demmig

#### Copyright und Urheberrechte:

Die durch die dpunkt.verlag GmbH vertriebenen digitalen Inhalte sind urheberrechtlich geschützt. Der Nutzer verpflichtet sich, die Urheberrechte anzuerkennen und einzuhalten. Es werden keine Urheber-, Nutzungs- und sonstigen Schutzrechte an den Inhalten auf den Nutzer übertragen. Der Nutzer ist nur berechtigt, den abgerufenen Inhalt zu eigenen Zwecken zu nutzen. Er ist nicht berechtigt, den Inhalt im Internet, in Intranets, in Extranets oder sonst wie Dritten zur Verwertung zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und eine gewerbliche Vervielfältigung der Inhalte wird ausdrücklich ausgeschlossen. Der Nutzer darf Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte im abgerufenen Inhalt nicht entfernen.

2. AUFLAGE

---

# Das DevOps-Handbuch

*Teams, Tools und Infrastrukturen erfolgreich  
umgestalten*

***Gene Kim, Jez Humble, Patrick  
Debois, John Willis, Nicole Forsgren***

*Deutsche Übersetzung von Thomas Demmig*

**O'REILLY®**

Gene Kim, Jez Humble, Patrick Debois, John Willis, Nicole Forsgren

Lektorat: Alexandra Follenius

Übersetzung: Thomas Demmig

Korrektorat: Sibylle Feldmann, [www.richtiger-text.de](http://www.richtiger-text.de)

Satz: III-Satz, [www.drei-satz.de](http://www.drei-satz.de)

Herstellung: Stefanie Weidner, Frank Heidt

Umschlaggestaltung: Karen Montgomery, Michael Oréal, [www.oreal.de](http://www.oreal.de)

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-199-8

PDF 978-3-96010-696-8

ePub 978-3-96010-697-5

mobi 978-3-96010-698-2

2. Auflage 2022

Translation Copyright für die deutschsprachige Ausgabe © 2022 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Authorized German translation of the English original »The DevOps Handbook, Second Edition«

Copyright © 2021 by Gene Kim, Matthew »Jez« Humble, Patrick Debois, John Willis, and Nicole Forsgren, ISBN 9781950508402. This translation is published and sold by permission of IT Revolution Press LLC, which owns or controls all rights to publish and sell the same.

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

*Hinweis:*

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die

Einschweißfolie.



*Schreiben Sie uns:*

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: [komentar@oreilly.de](mailto:komentar@oreilly.de).

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autoren noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

---

# Inhalt

**Anmerkung des Verlegers zur zweiten Auflage**

**Vorwort zur zweiten Auflage: Nicole Forsgren**

**Vorwort zur ersten Auflage: John Allspaw**

**Einleitung**

**Eine Einführung in das DevOps-Handbuch**

---

## **Teil I: Die Drei Wege**

### **1 Agile, Continuous Delivery und die Drei Wege**

Neue Fallstudie: Flughöhe erreicht: die DevOps-Reise von American Airlines (Teil 1) (2020)

### **2 Der Erste Weg: Die Prinzipien des Flow**

Neue Fallstudie: Flow und Constraint Management im Gesundheitswesen (2021)

### **3 Der Zweite Weg: Die Prinzipien des Feedbacks**

Neue Fallstudie: Die Andon-Cord bei Excella ziehen (2018)

### **4 Der Dritte Weg: Die Prinzipien des kontinuierlichen Lernens und Experimentierens**

Neue Fallstudie: Die Geschichte der Bell Labs (1925)

---

## **Teil II: Wie man beginnt**

### **5 Die Auswahl der ersten Wertkette**

Fallstudie: Die DevOps-Transformation bei Nordstrom (2014-2015)

Neue Fallstudie: Kessel Run: die Brownfield-Transformation bei Luftbetankungssystemen (2020)

Neue Fallstudie: DevOps auf das gesamte Business ausdehnen: die DevOps-Reise von American Airlines (Teil 2) (2020)

Neue Fallstudie: Die Ökonomie vor dem Ruin bewahren (mit einem HyperScale PaaS) bei HMRC (2020)

### **6 Die Arbeit in der Wertkette verstehen, sie sichtbar machen und verbreiten**

Fallstudie: Nordstroms Erfahrungen mit dem Value-Stream-Mapping (2015)

Fallstudie: Operation InVersion bei LinkedIn (2011)

### **7 Organisation und Architektur anhand von Conways Gesetz entwerfen**

Fallstudie: Conways Gesetz bei Etsy (2015)  
Fallstudie: API Enablement bei Target (2015)

## **8 Operations in die tägliche Entwicklungsarbeit integrieren**

Fallstudie: Big Fish Games (2014)  
Neue Fallstudie: Besseres Arbeiten bei Nationwide  
Building Society (2020)

---

# **Teil III: Der Erste Weg: Die technischen Praktiken des Flow**

## **9 Die Grundlagen für unsere Deployment- Pipeline legen**

Die Enterprise Data Warehouse Story (2009)  
Neue Fallstudie: Wie eine Hotelkette 30 Milliarden  
Dollar Umsatz in Containern generierte (2020)

## **10 Schnelles und zuverlässiges automatisiertes Testen ermöglichen**

Die Geschichte von Googles Webserver (2005)

## **11 Continuous Integration ermöglichen und umsetzen**

Die Firmware des HP LaserJet (2014)  
Fallstudie: Continuous Integration bei Bazaarvoice  
(2012)

## **12 Releases automatisieren und ihr Risiko reduzieren**

Fallstudie: Tägliche Deployments bei CSG  
International (2013)

Fallstudie: Etsy – Self-Service Developer Deployment, ein Beispiel für Continuous Deployment (2014)

Fallstudie: Blue-Green-Deployment für Point-of-Sale-Systeme bei Dixons Retail (2008)

Fallstudie: Dark Launch für den Facebook-Chat (2008)

Neue Fallstudie: Win-win für Dev und Ops bei CSG (2016)

### **13 Eine Architektur für risikoarme Releases**

Fallstudie: Evolutionäre Architektur bei Amazon (2002)

Fallstudie: Strangler-Muster bei Blackboard Learn (2011)

---

## **Teil IV: Der Zweite Weg: Die technischen Praktiken des Feedbacks**

### **14 Telemetriedaten erzeugen, um Probleme zu erkennen und zu beheben**

Fallstudie: DevOps-Transformation bei Etsy (2012)

Fallstudie: Self-Service-Metriken bei LinkedIn erstellen (2011)

### **15 Telemetriedaten analysieren, um Probleme besser vorherzusehen und Ziele zu erreichen**

Fallstudie: Telemetrie bei Netflix (2012)

Fallstudie: Automatisch skalierende Kapazität bei Netflix (2012)

Fallstudie: Fortgeschrittene Anomalie-Erkennung (2014)

## **16 Feedback ermöglichen, sodass Entwicklung und Operations Code sicher deployen können**

Fallstudie: Right Media (2006)

Fallstudie: Launch und Hand-off Readiness Review bei Google (2010)

## **17 Hypothesengetriebene Entwicklung und A/B-Tests in die tägliche Arbeit integrieren**

Fallstudie: Hypothesengetriebene Entwicklung bei Intuit, Inc. (2012)

Fallstudie: Doppeltes Umsatzwachstum durch schnelle Release-Zyklen und Experimente bei Yahoo! Answers (2010)

## **18 Review- und Koordinationsprozesse schaffen, um die Qualität der aktuellen Arbeit zu verbessern**

Fallstudie: Peer Review bei GitHub (2011)

Neue Fallstudie: Vom Sechs-Augen-Prinzip zum Releasen im großen Maßstab bei Adidas (2020)

Fallstudie: Code-Reviews bei Google (2010)

Fallstudie: Pair Programming ersetzt einen nicht funktionierenden Code-Review-Prozess bei Pivotal Labs (2011)

---

# **Teil V: Der Dritte Weg: Die technischen Praktiken des**

# **fortlaufenden Lernens und Experimentierens**

## **19 Lernen ermöglichen und Erfahrungen in die tägliche Arbeit einbringen**

Fallstudie: AWS US-East und Netflix (2011)

Neue Fallstudie: Einen Ausfall bei CSG in eine großartige Gelegenheit zum Lernen verwandeln (2021)

## **20 Lokale Erkenntnisse in globale Verbesserungen verwandeln**

Fallstudie: Einen neuen Technologie-Stack bei Etsy standardisieren (2010)

Neue Fallstudie: Crowdsourcing Technology Governance bei Target (2018)

## **21 Zeit für das firmenweite Lernen und für Verbesserungen reservieren**

Fallstudie: 30-Tage-Challenge bei Target (2015)

Fallstudie: Interne Technologie-Konferenzen bei Nationwide Insurance, Capital One und Target (2014)

---

## **Teil VI: Die technischen Praktiken zum Integrieren von Information Security, Änderungsmanagement und Compliance**

## **22 Information Security als Aufgabe für jeden Mitarbeiter an jedem Tag**

Fallstudie: Statische Sicherheitstests bei Twitter (2009)

Fallstudie: Compliance-Automatisierung durch 18F für die amerikanische Bundesverwaltung mit Compliance Masonry (2016)

Fallstudie: Instrumentieren der Umgebung bei Etsy (2010)

Neue Fallstudie: Shift Left für die Sicherheit bei Fannie Mae (2020)

## **23 Die Deployment-Pipeline schützen**

Fallstudie: Automatische Infrastrukturänderungen als Standardänderungen bei Salesforce.com (2012)

Fallstudie: PCI Compliance und ein abschreckendes Beispiel der Segregation of Duty bei Etsy (2014)

Neue Fallstudie: Biz- und Tech-Partnerschaft für zehn »No Fear Releases« pro Tag bei Capital One (2020)

Fallstudie: Compliance in regulierten Umgebungen belegen (2015)

Fallstudie: Verwenden von Produktiv-Telemetriedaten für Geldautomatensysteme

## **24 Ein Aufruf zum Handeln**

## **25 Nachwort zur zweiten Auflage**

---

# **Anhänge: Zusatzmaterial**

## **A Die Konvergenz von DevOps**

- B Theory of Constraints und zentrale chronische Konflikte**
- C Tabellarische Form der Abwärtsspirale**
- D Die Gefahren von Übergaben und Queues**
- E Mythen der Arbeitssicherheit**
- F Die Toyota-Andon-Cord**
- G COTS-Software**
- H Post-Mortem-Meetings (Retrospektiven)**
- I Die Simian Army**
- J Transparent Uptime**
- K Literatur**
- L Danksagungen**
- Index**

---

# Anmerkung des Verlegers zur zweiten Auflage

## Einfluss der ersten Auflage

Seit der ersten Veröffentlichung des *DevOps-Handbuchs* haben Daten aus den *State of DevOps Reports* und andere Forschungsergebnisse immer wieder gezeigt, dass DevOps den Time to Value für Firmen verbessert sowie die Produktivität und das Wohlbefinden der Mitarbeiter steigert. DevOps hilft auch dabei, flexible und agile Unternehmen zu schaffen, die sich an außergewöhnliche Veränderungen anpassen können, wie das in der seit 2020 andauernden COVID-19-Pandemie zu sehen war.

»Ich denke, 2020 hat gezeigt, was Technologie in Zeiten außerordentlicher Krisen leisten kann«, schreibt Gene Kim in seinem Artikel »State of DevOps: 2020 and Beyond«. »Die Krise diente als Katalysator für rapide Veränderungen. Und ich bin dankbar, dass wir sie bewältigen konnten.«<sup>1</sup>

Eine der Ideen von DevOps und dem *DevOps-Handbuch* ist, dass es die Arbeitspferde und nicht die Einhörner der Firmen und der Technologie-Welt vorstellt – und auch dafür geschrieben ist. DevOps war nie – und ist es auch weiterhin

nicht - nur für Technologie-Riesen wie den FAANGs (Facebook, Amazon, Apple, Netflix und Google) oder Start-ups effektiv. Dieses Buch und die DevOps-Community im Ganzen haben immer wieder gezeigt, dass DevOps-Praktiken und -Prozesse selbst von Altsystemen durchzogene Organisationen in alten »Arbeitspferd«-Unternehmen in flexible Technologie-Organisationen verwandeln können.

Heute wird deutlicher als je zuvor, dass jede Firma eine Technologie-Firma und jeder Leader ein Technologie-Leader ist. Technologie kann nicht nur nicht länger ignoriert oder in den Keller verbannt werden - sie ist auch als vitaler Teil der gesamten Strategemaßnahmen der Firma zu berücksichtigen.

## **Änderungen zur zweiten Auflage**

In dieser erweiterten Auflage des *DevOps-Handbuchs* haben die Autorin und die Autoren den Haupttext dort aktualisiert, wo neue Forschungen, Ergebnisse und Erfahrungen unser Verständnis von DevOps und seinen Einsatz in der Branche weiterentwickelt und neu geformt haben. Zudem freuen wir uns darüber, die renommierte Forscherin Dr. Nicole Forsgren als Koautorin gewonnen zu haben, die dabei half, den Text durch neue Untersuchungen und weitere Metriken zu aktualisieren.

### **Continuous Learning**

Wir haben ein paar zusätzliche Einsichten und Ressourcen mit aufgenommen, die uns erst nach dem

Erscheinen der ersten Auflage aufgefallen sind oder die erst danach veröffentlicht wurden. Diese »Continuous Learning«-Abschnitte sind im Buch wie hier hervorgehoben und enthalten neue Daten und zusätzliche Ressourcen, Werkzeuge und Techniken, die Sie auf Ihrer DevOps-Reise einsetzen können.

Außerdem haben wir das Buch durch zusätzliche Fallstudien ergänzt, um zu zeigen, wie weit sich DevOps in allen Branchen verbreitet hat – insbesondere wie es über die IT-Abteilungen hinaus bis in die C-Ebene vorgedrungen ist. Zusätzlich haben wir am Ende jeder Fallstudie ein oder zwei wichtige Schlussfolgerungen angefügt, die die wichtigsten (aber nicht die einzigen) Erkenntnisse daraus zusammenfassen. Und schließlich haben wir die Zusammenfassung am Ende jedes Buchteils durch neue Ressourcen ergänzt, mit denen Sie Ihre Reise fortführen können.

## **Wie geht es mit DevOps und dem Softwarezeitalter weiter?**

Wenn uns die letzten fünf Jahre etwas gelehrt haben, dann, dass Technologie immer wichtiger wird und dass wir viel erreichen können, wenn IT und Business offen und ehrlich miteinander reden – so wie DevOps es ermöglicht.

Vielleicht wird das am deutlichsten durch die rapiden Veränderungen, die aufgrund der COVID-19-Pandemie seit dem Jahr 2020 erforderlich sind. Mithilfe von DevOps konnten Organisationen Technologie nutzen, um internen und externen Kunden während Zeiten großer Verwerfungen Services anzubieten. Diese großen,

komplexen Organisationen, die für ihre Unfähigkeit bekannt waren, sich zu verändern und sich schnell anzupassen, hatten plötzlich keine andere Wahl mehr.

American Airlines konnte sich die laufende DevOps-Transformation ebenfalls zunutze machen, um schnell große Gewinne einzufahren, wie Sie in den [Kapiteln 1](#) und [5](#) lesen werden.

Dr. Chris Strear berichtet in [Kapitel 2](#) von seinen Erfahrungen mit der Theory of Constraints, um die Abläufe in Krankenhäusern zu optimieren.

Im Jahr 2020 konnte Nationwide Building Society - die weltgrößte Bausparkasse - dank ihrer aktuellen DevOps-Transformation innerhalb von Wochen auf Kundenanforderungen reagieren, statt wie sonst üblich in Jahren zu agieren. Mehr über ihre Erfahrungen lesen Sie in [Kapitel 8](#).

Bei der erfolgreichen Transformation hin zu zukunftssträchtigen Arbeitsweisen spielt die Technologie eine wichtige Rolle, aber Business Leadership muss die Führung übernehmen. Den Flaschenhals bilden heutzutage nicht mehr technische Vorgehensweisen (auch wenn es diese Engpässe weiterhin gibt), die größte Herausforderung - damit auch die schwierigste Anforderung - besteht darin, die Firmenleitung mit ins Boot zu holen. Transformation muss gemeinsam durch Business und Technologie umgesetzt werden, und die hier vorgestellten Theorien können diese Veränderung unterstützen.

Das Unternehmen kann nicht mehr länger in binären Denkmustern agieren - Top-down oder rein technologiegetrieben. Wir müssen eine echte Zusammenarbeit erreichen. Neunzig Prozent dieser Arbeit

dreht sich darum, die richtigen Menschen mit einzubeziehen und engagiert auf einen sinnvollen Weg zu schicken. Beginnen Sie damit, kann uns die sich daraus ergebende Motivation in die Zukunft tragen.

*IT Revolution  
Portland, OR Juni 2021*

---

# Vorwort zur zweiten Auflage

Seit der Veröffentlichung der ersten Auflage des DevOps-Handbuchs sind fünf Jahre vergangen. Es hat sich viel verändert, aber es ist auch vieles gleich geblieben. Manche unserer Werkzeuge und Technologien sind aus der Mode gekommen, andere überhaupt nicht mehr vorhanden, aber das sollte Sie als Leserin oder Leser nicht abschrecken. Auch wenn sich die technologische Landschaft verändert hat, bleiben die diesem Buch zugrunde liegenden Prinzipien genauso relevant wie immer.

Tatsächlich ist der Bedarf an DevOps heutzutage sogar noch gewachsen, wenn Organisationen schneller, sicherer und zuverlässiger Werte an ihre Kunden und Anwenderinnen liefern müssen. Dazu müssen sie ihre internen Prozesse transformieren und mithilfe von Technologie Wert bieten - unter Rückgriff auf DevOps-Praktiken. Das gilt für Organisationen auf der ganzen Welt und in allen Branchen.

In den letzten Jahren habe ich Untersuchungen für die jährlichen *State of DevOps Reports* geleitet (zuerst mit Puppet, später mit DORA und Google) - zusammen mit den

Koautoren Jez Humble und Gene Kim. Die Forschung hat gezeigt, dass viele der in diesem Buch beschriebenen Praktiken die Ergebnisse wie Geschwindigkeit und Stabilität von Software-Releases verbessern – ebenso wie sie zur Leistung von Organisationen beitragen, unter anderem zu Profitabilität, Produktivität, Kundenzufriedenheit, Effektivität und Effizienz.

Für die zweite Auflage des DevOps-Handbuchs haben wir den Text mit aktualisierten Daten überarbeitet, die auf neuesten Forschungen und Best Practices basieren, und neue Fallstudien mit aufgenommen, um noch mehr Geschichten davon zu erzählen, wie Transformationen »an der Basis« aussehen. Vielen Dank, dass Sie uns auf dieser Reise durch die fortwährende Verbesserung begleiten.

*Nicole Forsgren, PhD  
Partner bei Microsoft Research 2021*

---

# Vorwort zur ersten Auflage

In der Vergangenheit gab es in vielen Bereichen des Ingenieurwesens auf die eine oder andere Weise merklichen Fortschritt und ein wachsendes Verständnis der eigenen Arbeit. Es gibt in vielen Ingenieurdisziplinen universitäre Weiterbildung und Organisationen, die Expertenhilfe bieten (Bauwesen, Maschinenbau, Elektrotechnik, Nukleartechnik usw). Tatsächlich braucht die moderne Gesellschaft auch alle Arten von Ingenieuren, um die Vorteile der interdisziplinären Zusammenarbeit zu erkennen und diese umzusetzen.

Denken Sie an die Konstruktion eines modernen Autos. Wo endet die Arbeit eines Maschinenbauers, und wo beginnt die eines Elektrotechnikers? Wo (und wie und wann) sollte jemand mit Aerodynamik-Kenntnissen (der sicherlich eine ziemlich genaue Vorstellung davon hat, wie die Fenster aussehen sollten, wie groß sie zu sein haben und wo sie sitzen sollten) mit einem Experten für die Sitzergonomik zusammenarbeiten? Was ist mit dem chemischen Einfluss des Treibstoffs und des Öls auf die Materialien von Motor und Schläuchen über die gesamte Lebenszeit des Autos? Es gibt zum Design eines Autos noch viel mehr Fragen, die man stellen kann, aber das Ergebnis ist immer das gleiche: Erfolg stellt sich in modernen technischen Projekten nur ein, wenn mehrere Sichtweisen und unterschiedliches Expertenwissen zusammenkommen.

Damit sich ein technischer Bereich weiterentwickelt, muss ein Punkt erreicht werden, an dem man sich Gedanken über den bisherigen Weg macht, diesen von mehreren Seiten beleuchtet und das Ganze so zusammenfasst, dass es für die Vorstellungen der Community zur Zukunft hilfreich ist.

Dieses Buch bietet diese Zusammenfassung und sollte als grundlegende Sammlung von Sichtweisen auf das (zugegebenermaßen sich noch entwickelnde und wachsende) Feld von Softwareentwicklung und Operations gesehen werden.

Egal in welcher Branche Sie arbeiten oder welches Produkt oder welchen Service Ihre Firma bereitstellt – diese Art des Denkens ist für das Überleben jedes Business und Technology Leader außerordentlich wichtig.

*John Allspaw, CTO, Etsy  
Brooklyn, NY, August 2016*

---

# Einleitung

*Aha!*

Das *DevOps-Handbuch* ist nicht von heute auf morgen entstanden – es begann im Februar 2011 mit wöchentlichen Skype-Telefonaten zwischen den Autoren und der Vision, dem damals noch unvollendeten Buch *Projekt Phoenix: Der Roman über IT und DevOps – Neue Erfolgsstrategien für Ihre Firma* ein präskriptives Handbuch zur Seite zu stellen.

Über fünf Jahre und mehr als zweitausend Arbeitsstunden später ist das *DevOps-Handbuch* endlich fertig. Ein außerordentlich langwieriger Prozess, der aber sehr lohnenswert und lehrreich war und den Rahmen viel weiter spannte, als wir uns das ursprünglich vorgestellt hatten. Alle Autoren eint, dass sie an DevOps als außerordentlich wichtiges Konzept glauben, weil sie in ihrem Berufsleben einen »Aha-Moment« hatten, den unsere Leser sicherlich interessiert.

## **Gene Kim**

Ich hatte die Möglichkeit, seit 1999 erfolgreiche Technologie-Firmen untersuchen zu können, und eine der ersten Erkenntnisse war, dass die bereichsübergreifende Zusammenarbeit der unterschiedlichen Gruppen – IT Operations, Information Security und Development – für

den Erfolg entscheidend ist. Ich erinnere mich vor allem an das erste Mal, als ich die Abwärtsspirale beobachten konnte, die entstand, weil diese Gruppen unterschiedliche Ziele verfolgten.

In 2006 hatte ich die Gelegenheit, für eine Woche eine Gruppe zu begleiten, die die outgesourcten IT Operations eines großen Flugbuchungsdienstleisters betreute. Mir wurden die negativen Folgen der umfangreichen jährlichen Software-Releases beschrieben, die jedes Mal zu großem Chaos und Unterbrechungen für den Outsourcer, aber auch für die Kunden führen würden. Es gäbe wegen der für den Kunden bemerkbaren Ausfälle Strafen aufgrund nicht eingehaltener SLAs (*Service Level Agreement*), der Gewinn bräche ein, und daher würden die talentiertesten und erfahrensten Mitarbeiter entlassen werden. Es müssten mehr ungeplante Aufgaben erledigt und überall Feuer gelöscht werden. Die verbleibende Crew hätte noch weniger Zeit, den stetig wachsenden Service Request Backlog der Kunden abzuarbeiten. Die Verträge würden überhaupt nur durch den massiven Einsatz des mittleren Managements aufrechterhalten werden, und alle hätten Angst, dass sie in drei Jahren bei einer Neuausschreibung sowieso nicht mehr zum Zuge kämen.

Das Gefühl der Hoffnungslosigkeit und Sinnlosigkeit der Arbeit war für mich der Startschuss für meinen moralischen Kreuzzug. Development schien immer als strategisch wichtig angesehen zu werden, während IT Operations nur taktisch betrachtet wurde - etwas, das man gern wegdelegierte oder gleich ganz auslagerte, nur um es fünf Jahre später in einer noch schlechteren Verfassung wiederzubekommen.

Während all der Jahre war den meisten von uns bewusst, dass es einen besseren Weg geben musste. Ich erinnere

mich an die Vorträge der Velocity Conference im Jahr 2009, in denen die erstaunlichen Ergebnisse vorgestellt wurden, die aus der Architektur, den Praktiken und kulturellen Normen entstanden, die wir als DevOps kennen. Ich war total begeistert, weil hier der bessere Weg, nach dem wir alle suchten, deutlich aufgezeigt wurde. Dieses Wissen zu verbreiten, war Teil meiner persönlichen Motivation, am *Projekt Phoenix* mitzuarbeiten. Sie können sich vorstellen, wie glücklich ich darüber war, dass viele Menschen durch das Buch ihre eigenen Aha-Momente hatten.

## **Jez Humble**

Mein Aha-Moment mit DevOps erlebte ich bei einem Start-up im Jahr 2000 - in meinem ersten Job, den ich nach dem Studium annahm. Eine Zeit lang war ich einer von zwei Technikmitarbeitern. Ich machte alles: Netzwerk, Programmieren, Support, Systemverwaltung. Wir deployten Software in die Produktivumgebung, indem wir sie per FTP direkt von unseren Workstations hochluden.

2004 erhielt ich dann einen Job bei ThoughtWorks - einer Consulting-Firma, in der ich als Erstes in einem Projekt zusammen mit ungefähr 70 Leuten arbeiten sollte. Ich gehörte zu einem Team aus acht Entwicklern, deren einzige Aufgabe das Deployen unserer Software in eine der Produktivumgebung ähnliche Umgebung war. Zu Beginn war das sehr stressig. Aber im Laufe der Zeit schafften wir es, von einem manuellen Deployment, das zwei Wochen brauchte, zu einem automatisierten zu gelangen, das nur eine Stunde dauerte und bei dem der Wechsel zwischen alter und neuer Umgebung in Millisekunden vonstattenging - während der normalen Arbeitszeit und mithilfe des Blue-Green-Deployment-Musters.

Dieses Projekt war Inspirationsquelle für viele der Ideen in *Continuous Delivery* und in diesem Buch. Mich und viele andere, die in diesem Bereich unterwegs sind, treibt das Wissen an, dass wir es trotz aller Einschränkungen immer noch besser machen können und dass ich den Menschen auf ihrer Reise dorthin helfen möchte.

## **Patrick Debois**

Bei mir waren es mehrere Momente. 2007 arbeitete ich in einem Data-Center-Migration-Projekt mit ein paar Agile-Teams zusammen. Ich war neidisch, dass sie so produktiv waren und in kurzer Zeit so viel erledigen konnten.

In meinem nächsten Projekt begann ich, in Operations mit Kanban zu experimentieren, und ich beobachtete, wie sich die Teamdynamik änderte. Später stellte ich auf der Agile Toronto Conference 2008 mein IEEE-Paper dazu vor, hatte aber das Gefühl, dass es in der Agile-Community nicht groß wahrgenommen wurde. Wir setzten eine Agile-System-Administration-Gruppe auf, aber ich achtete nicht auf die menschliche Seite des Ganzen.

Nachdem ich auf der Velocity Conference 2009 die Präsentation »10 Deploys per Day« von John Allspaw und Paul Hammond gesehen hatte, war ich davon überzeugt, dass auch andere wie ich dachten. Also entschied ich mich dazu, die ersten DevOpsDays zu organisieren, und prägte damit unabsichtlich den Begriff DevOps.

Die Energie auf diesem Event war einmalig und ansteckend. Als die Menschen anfangen, sich bei mir dafür zu bedanken, dass ihr Leben nun besser geworden sei, verstand ich den Einfluss, den das Ganze hatte. Seitdem habe ich nicht mehr damit aufgehört, DevOps anzupreisen.

## John Willis

2008 hatte ich gerade eine Consulting-Firma verkauft, die auf Beratung im Umfeld von großen Legacy-IT-Operations rund um Configuration Management und Monitoring spezialisiert war (Tivoli). Ich traf Luke Kanies (den Begründer von Puppet Labs), der auf einer Open-Source-Konferenz von O'Reilly einen Vortrag über *Configuration Management* (CM) hielt.

Zuerst lungerte ich nur hinten im Vortragsraum herum, schlug die Zeit tot und dachte: »Was kann mir dieser Zwanzigjährige schon über Configuration Management erzählen?« Schließlich hatte ich schon mein ganzes Leben bei ein paar der größten Unternehmen der Welt gearbeitet und ihnen dabei geholfen, die Architektur für ihr CM und andere Operations-Management-Lösungen aufzubauen. Aber nach fünf Minuten setzte ich mich ganz nach vorne und erkannte, dass alles, was ich in den letzten 20 Jahren gemacht hatte, falsch gewesen war. Luke beschrieb das, was ich heute als CM der zweiten Generation bezeichne.

Nach seiner Session hatte ich die Gelegenheit, mich mit ihm auf einen Kaffee zusammensetzen. Von dem, was wir heute als *Infrastructure as Code* bezeichnen, war ich vollkommen überzeugt. Aber bei unserem Gespräch ging Luke noch weiter und erläuterte mir seine Ideen. Er glaubte, dass sich Operations mehr wie Softwareentwicklung verhalten müsse. Sie müssten ihre Konfigurationen unter Versionsverwaltung stellen und für ihren Workflow CI/CD-Delivery-Muster übernehmen. Ich war damals noch der Oldschool-IT-Operations-Mensch und antwortete wohl mit so etwas wie: »Diese Idee wird so wenig Erfolg haben wie Led Zeppelin mit ihrem Folk-Kram.« (Ich lag so was von falsch!)

Etwa ein Jahr später besuchte ich auf einer anderen O'Reilly-Konferenz (Velocity) einen Vortrag von Andrew Clay Shafer über Agile Infrastructure. Dabei zeigte Andrew dieses berühmte Bild einer Mauer zwischen Entwicklern und Operations, bei der die Arbeit immer nur hinübergeworfen wird. Dieses Bild bezeichnete er als »Wall of Confusion«. Die Ideen in diesem Vortrag kodifizierten das, was Luke mir ein Jahr zuvor zu erklären versuchte, und mir ging ein Licht auf. Im selben Jahr war ich der einzige Amerikaner, der zu den ersten DevOpsDays in Gent eingeladen wurde. Als dieses Event vorbei war, floss DevOps durch meine Adern.

## **DevOps-Mythen**

Alle Autoren dieses Buchs hatten ganz klar die gleiche Erscheinung, auch wenn sie aus verschiedenen Richtungen kamen. Aber es gibt eine unglaubliche Menge an Belegen dafür, dass die beschriebenen Probleme so gut wie überall auftauchen und dass sich die Lösungen, die mit DevOps verbunden sind, nahezu universell einsetzen lassen.

Dieses Buch will beschreiben, wie Sie die Transformationen durch DevOps, an denen wir beteiligt waren oder die wir beobachten konnten, selbst umsetzen können, es will aber auch mit vielen Mythen aufräumen, die vorzugeben versuchen, warum DevOps in bestimmten Situationen nicht funktionieren wird. Ein paar der bekanntesten Mythen, die wir über DevOps gehört haben, sind:

### **Mythos: DevOps ist nur für Start-ups**

DevOps-Praktiken entstanden zwar in Internet-»Unicorn«-Firmen wie Google, Amazon, Netflix und Etsy, aber alle waren an einem das Geschäft gefährdenden Punkt

angelangt - aufgrund von Problemen, die eher mit den klassischen »Horses«-Firmen verbunden sind: hochriskante Code-Releases, die zu katastrophalen Fehlschlägen zu werden drohten, fehlende Möglichkeiten, Features schnell genug veröffentlichen zu können, um so der Konkurrenz zuvorzukommen, Compliance-Sorgen, die Unfähigkeit, skalieren zu können, riesiges Misstrauen zwischen Entwicklung und Operations usw.

Aber alle diese Firmen schafften es, ihre Architektur, die technischen Praktiken und ihre Kultur so umzustellen, dass diese erstaunlichen Ergebnisse entstanden, die wir mit DevOps verbinden. Wie der Information Security Executive Dr. Branden Williams gern scherzt: »Reden wir nicht mehr über DevOps-Einhörner oder -Pferde, sondern nur noch über Vollblüter oder Pferde, die zum Abdecker müssen.«<sup>1</sup>

## **Mythos: DevOps ersetzt Agile**

DevOps-Prinzipien und -Praktiken sind kompatibel zu Agile, und viele haben das Gefühl, dass es sich bei DevOps um eine logische Fortsetzung der Agile-Reise handelt, die 2001 ihren Anfang nahm. Agile dient häufig als effektiver Türöffner für DevOps, weil es sich auf kleine Teams fokussiert, die den Kunden kontinuierlich qualitativ hochwertigen Code liefern.

Viele DevOps-Praktiken entwickeln sich daraus, dass wir unsere Arbeit über das Ziel des »potenziell auslieferbaren Codes« am Ende jeder Iteration hinaus fortführen und dafür sorgen, dass er sich stets in einem auslieferbaren Zustand befindet, den Entwickler täglich einchecken und mit dem wir unsere Features in produktivähnlichen Umgebungen präsentieren.

## **Mythos: DevOps ist nicht kompatibel zu ITIL**

Viele sehen DevOps als Rückschritt gegenüber ITIL oder ITSM (*IT Service Management*), das ursprünglich 1989 veröffentlicht wurde. ITIL hat viele Generationen von Ops-Mitarbeitern beeinflusst, einen der Autoren eingeschlossen, und ist eine stetig weiterentwickelte Praxisbibliothek, mit der die Prozesse und Vorgehensweisen erstklassiger IT Operations kodifiziert werden sollen - einschließlich Servicestrategie, Design und Support.

DevOps-Praktiken können kompatibel zum ITIL-Prozess gestaltet werden. Um die mit DevOps in Verbindung gebrachten kürzeren Durchlaufzeiten und höheren Deployment-Frequenzen zu unterstützen, müssen viele Teile von ITIL vollständig automatisiert werden, was eine Reihe von Problemen rund um das Konfigurations- und Release-Management löst (zum Beispiel werden so die Konfigurationsmanagement-Datenbank und die Softwarebibliotheken aktuell gehalten). Und weil zu DevOps auch das schnelle Erkennen von und Reagieren auf Serviceprobleme gehören, bleiben die ITIL-Bereiche Servicedesign sowie Störungs- und Problemmanagement so wichtig wie zuvor.

## **Mythos: DevOps ist nicht kompatibel zu Information Security und Compliance**

Das Fehlen klassischer Steuerelemente (zum Beispiel der Segregation of Duty, eines Änderungsgenehmigungsprozesses oder manueller Security Reviews am Ende des Projekts) mag Experten für Information Security und Compliance in Panik versetzen.

Aber das heißt nicht, dass DevOps-Organisationen keine effektiven Steuerelemente besitzen. Statt Security- und