

# Devops

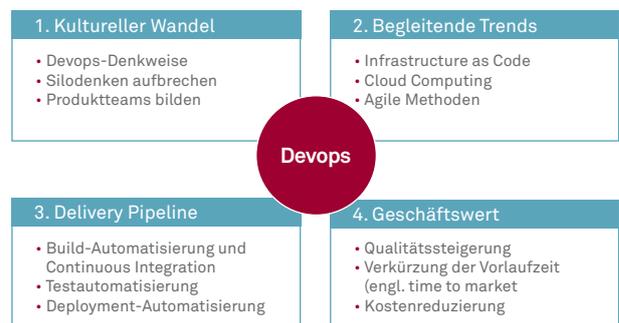
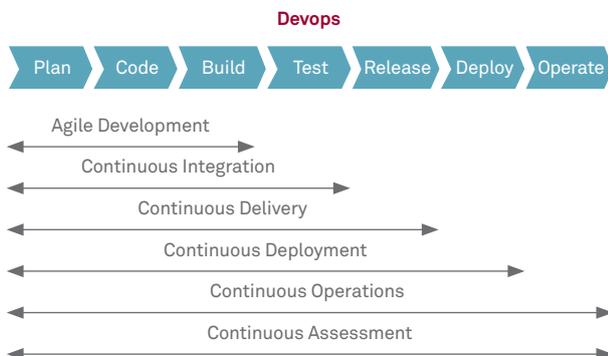
## Agile Softwareentwicklung konsequent zu Ende gedacht

Ein wesentliches Ziel der agilen Softwareentwicklung war schon immer, den Auftraggeber möglichst frühzeitig in die Entwicklung einzubinden. Denn frühes und häufiges Feedback soll sicherstellen, dass der Auftraggeber stets das Produkt mit dem für ihn größten Mehrwert erhält. Durch eine konsequente Umsetzung der Devops-Prinzipien lässt sich nicht nur ein nahezu kontinuierlicher Feedback-Prozess erreichen, sondern auch kürzere Releasezyklen bei gleichzeitig hoher Softwarequalität sicherstellen.

### Definition

Devops verzahnt sowohl buchstäblich als auch semantisch die Bereiche Development und Operations. Entwickler und Betriebler rücken zusammen und arbeiten schlussendlich am selben Tisch. Devops lässt sich deshalb vornehmlich als kulturelle Strömung verstehen. Ihr Ziel ist es, die Geschwindigkeit der Entwicklung, der Auslieferung sowie die Software-Qualität zu erhöhen, indem Silodenken aufgebrochen und das Schwarzer-Peter-Spiel (engl. blame game) abgeschafft werden.

Devops geht aber über reine Entwickler- und Betriebsteams hinaus. Es fördert vielmehr eine enge Zusammenarbeit aller am Produkt beteiligten Personen. Das führt zu einem organisatorischen Wandel im Unternehmen. Die operative Betriebsverantwortung liegt dann nicht mehr beim Ops-Team, sondern beim zu etablierenden, interdisziplinär besetzten Produktteam. Jedes Mitglied des Produktteams ist somit unmittelbar am Produkterfolg beteiligt und ebenso mitverantwortlich; frei nach dem Motto „you build it - you run it“.



### Referenzszenario

Ein Unternehmen möchte seinen Kunden qualitativ hochwertige neue Features auf Knopfdruck und mit geringem operativen Risiko zur Verfügung stellen. Dazu etabliert das Unternehmen eine disziplinarisch übergreifende Zusammenarbeit der verschiedenen Produktteams unter Einbeziehung aller Stakeholder für einen reibungsfreien Ablauf.

Jedes Produktteam steht in der Verantwortung für sein Produkt, von der ersten Zeile Code bis zum Betrieb. Um dieser Verantwortung gerecht zu werden, steht dem Team auf operativer Ebene eine ausgereifte Toolchain zur Verfügung. Das Team nutzt virtuelle Rechenressourcen in der Cloud, die es mittels Infrastructure as Code konfiguriert und so eine Unabhängigkeit von außen garantieren kann. Ein Buildserver baut, testet und rollt nach einer Codeänderung automatisiert Applikationen der Microservice-Architektur in Docker-Container aus. Eine hohe Testabdeckung stellt dabei eine hohe Produktqualität sicher. Außerdem erheben die Teams zahlreiche Metriken. So sind sie permanent über den Zustand der Komponenten im Bilde und können bei Bedarf gegensteuern.

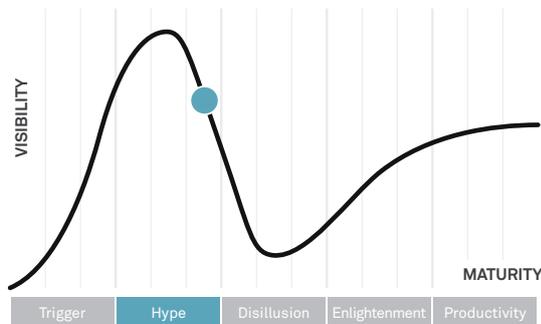
## Business Impact

In Zeiten der Digitalen Transformation und im Umfeld von Systems of Engagement ist es ein entscheidender Wettbewerbsvorteil, neue Features in immer kürzeren Intervallen an die Anwender zu verteilen. Durch Devops lassen sich die organisatorischen, methodischen und operativen Voraussetzungen für kurze Feature-Intervalle erfüllen.

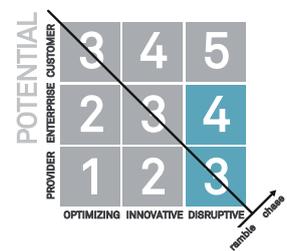
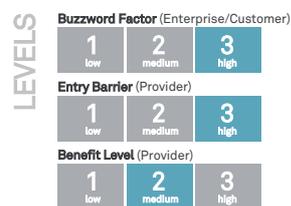
Die Einführung der Devops-Prinzipien bedeutet einen signifikanten Aufwand. So ist eine organisatorische Umstrukturierung nötig, um die Übernahme des Betriebs durch das interdisziplinäre Produktteam überhaupt zu ermöglichen. Unternehmen, welche an den klassischen Verfahren festhalten, laufen hingegen Gefahr, im Wettbewerb in Schwierigkeiten zu geraten, weil die Konkurrenz ihre Kosten, Qualität und Vorlaufzeit (engl. time to market) durch die Umsetzung von Devops-Prinzipien optimieren.

## Reifegrad

Devops entstand 2013 während einer Konferenz und wird mittlerweile auch bei namhaften Firmen im großen Stil produktiv umgesetzt. Auch in msg- und Kundenprojekten finden ausgewählte Ansätze, Methoden und Tools bereits Anwendung. Die größte Hürde besteht dabei auf organisatorischer Seite, etwa bei der Bildung der interdisziplinären Produktteams. Auf methodischer Seite stellt sich der Schnitt der Artefakte noch als Hürde dar.



Vorhaben	Tool
IDE	Eclipse oder IntelliJ IDEA
Virtualisierung und Container	VMWare, KVM, Docker
Projektplanung und Tickettool	Atlassian JIRA
Source Code Repository	Atlassian Stash, GIT, SVN
CI	Jenkins
Statische Code Analyse	SonarQube
Artifact Repository	Nexus, Artifactory
Buildsystem	Maven, Gradle
Tests	JUnit, Selenium, Mockito, JMeter
Security	OWASP ZAP Proxy, Burp Intruder
Provisionierung	Ansible, Puppet
Cloud	AWS, Azure, Openstack



## Marktübersicht

Viele Produkte schicken sich an, den Devops-Ansatz methodisch und operativ zu unterstützen und zu optimieren. Die Liste oben stellt verschiedenen Vorhaben passende Programme gegenüber.

## Alternativen

Um die Ziele zu erreichen, die mit dem Einsatz von Devops-Prinzipien verfolgt werden, gibt es kaum sinnvolle Alternativen. Denn die Devops-Prinzipien lassen sich auch losgelöst voneinander umsetzen. Diese Untermenge der Prinzipien führt dann in Umgebungen, die noch von konservativen Entwicklungs- und Release-Prozessen geprägt sind, bereits zu spürbaren Verbesserungen und produziert einen Mehrwert.

Pro	Contra
Kürzere Vorlaufzeit	Nicht einfach eine Tool-Sammlung
Verbesserte Produktivität	Anpassung des Mindsets und der Kultur erforderlich
Zuverlässige Releases	Kein Standard-Rezept – Einzelbetrachtung notwendig
Schnellere Feedback-Schleifen	Hohe Einstiegshürden
Erhöhte Kundenzufriedenheit	Kein Allheilmittel
Erhöhte Kosteneffizienz	Wandel vom Spezialisten zum T-Shaped-Professional