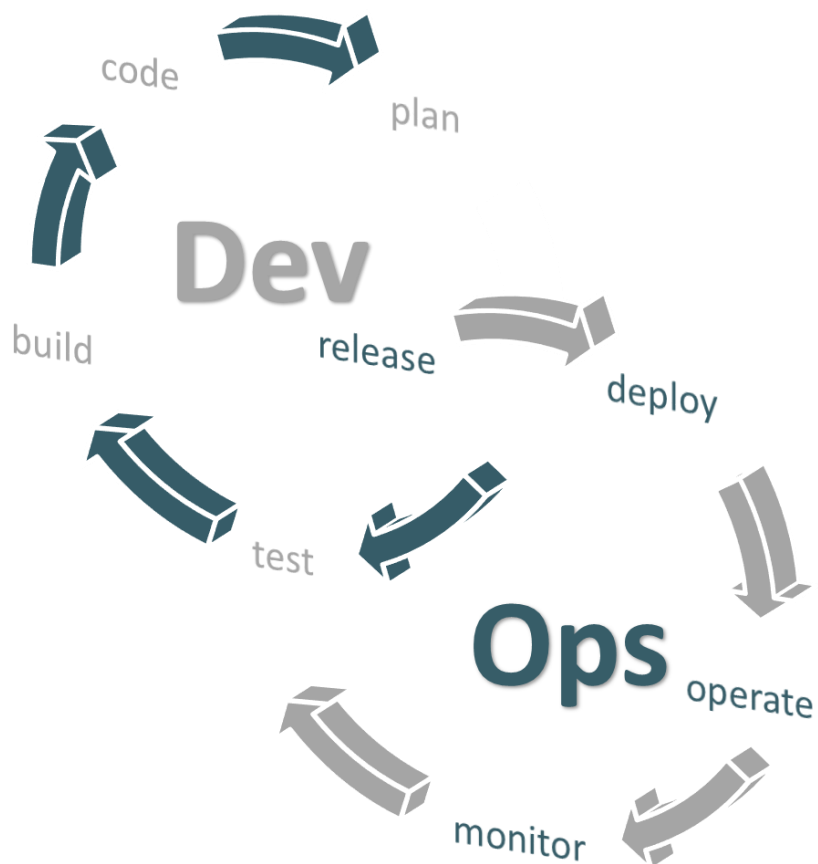


## DevOps als modernes IT-Outsourcing:

Agile Softwareentwicklung bis zum Ende gedacht



## DevOps als modernes IT-Outsourcing

### Inhalt

Inhaltsverzeichnis .....	2
<b>Executive Summary .....</b>	<b>3</b>
1 Einleitung .....	4
2 DevOps in der Praxis: Maßnahmen in Organisation und Technik .....	6
2.1 Neues Denken und gemeinsame Verantwortung als Schlüssel .....	6
2.2 Prozessoptimierung.....	6
2.3 Technische Hilfsmittel: DevOps-Tool-Kette .....	6
3 DevOps bei einem Outsourcing des IT-Betriebs an IT-Dienstleister.....	8
3.1 Ausgangspunkt .....	8
3.2 DevOps als Dienstleistung von Premium-IT-Dienstleistern.....	8
4 Praxiserfahrungen: Kostenvergleich klassischer Betrieb versus DevOps.....	10
5 Eignung für DevOps .....	13
6 Fazit.....	14
<b>Die noris network AG .....</b>	<b>15</b>

**Executive Summary**

DevOps ist die Ausdehnung der agilen Softwareentwicklung auf den operativen IT-Betrieb. Das Ziel: Die Kooperationsprobleme zwischen einer agil arbeitenden Software-Entwicklung (Development) und IT-Betrieb (Operations) sollen überwunden werden. Software für volatile Märkte oder Anwendungen mit besonders stark schwankendem Ressourcenbedarf (z. B. Webshops, Cloud-Computing) eignen sich besonders für eine agile Entwicklung und den darauf abgestimmten Betrieb. Premium-IT-Dienstleister wie noris network beschäftigen eigene Teams mit erfahrenen DevOps-Spezialisten und bieten Aufbau, Pflege und Weiterentwicklung hocheffizienter DevOps-Workflows als Dienstleistung an. Kontinuierliche Qualitätsprüfung und Auslieferung werden durch einen hohen Automationsgrad der Ressourcenzuteilung unterstützt. Dadurch lassen sich Ziele erreichen, die mit klassischen Prozessen (lang geplante Releases, Change Requests etc.) nicht vereinbar sind: Eine hohe Anzahl von Releases aus einer agilen Entwicklung (Time-to-Market), ein hohes Sicherheits- und Leistungsniveau (Stabilität, Verfügbarkeit) bei gleichzeitig niedrigeren Kosten im Betrieb (durch synchronisierte Tool-Ketten und Automatisierung).

**noris network AG**

## 1 Einleitung

### **Die DevOps-Bewegung: Agile Softwareentwicklung bis zum Ende gedacht**

Um die Jahrtausendwende begann der Siegeszug neuer Methoden in der Softwareentwicklung. Abgeleitet vom Lean-Gedanken in der industriellen Produktion sollten Prozesse in der Softwareentwicklung schlanker und flexibler werden. Die damals entwickelten Methoden wie Extreme Programming, Scrum oder Test-Driven Development werden heute als agile Softwareentwicklung zusammengefasst und in vielen Unternehmen gelebt.

Mit agilen Methoden wird vorrangig Software entwickelt, bei der die Interaktion mit Anwendern und die User Experience im Mittelpunkt stehen – Beispiel Cloud-Anwendungen oder Webshops. Der Grund: Hier sollen Ideen und Anforderungen möglichst schnell in Produkt-Features und somit in Produkt- und Wettbewerbsvorteile verwandelt werden.

Agile Softwareentwicklung ändert die Art, wie Software ausgerollt wird. Statt pro Jahr ein oder einige wenige, aber umfangreiche Releases herauszubringen, stellen die Entwickler mehrmals in der Woche oder gar mehrmals am Tag neue Builds zur Verfügung. Was aus Entwicklungssicht gut klingt, hat an anderen Stellen Konsequenzen: Agile Methoden mit erhöhter Frequenz von Build-Bereitstellungen nehmen keine Rücksicht auf die Notwendigkeit eines automatisierten, standardisierten IT-Betriebs mit verbindlichen Verfügbarkeiten und planbarer Ressourcennutzung. Die Folge: Erzeugen eingespielte Releases Probleme, kommt es zwischen Development- und Operations-Teams zum „Blame Game“, also zu gegenseitigen Schuldzuweisungen.

### **DevOps ist ein Kulturwandel**

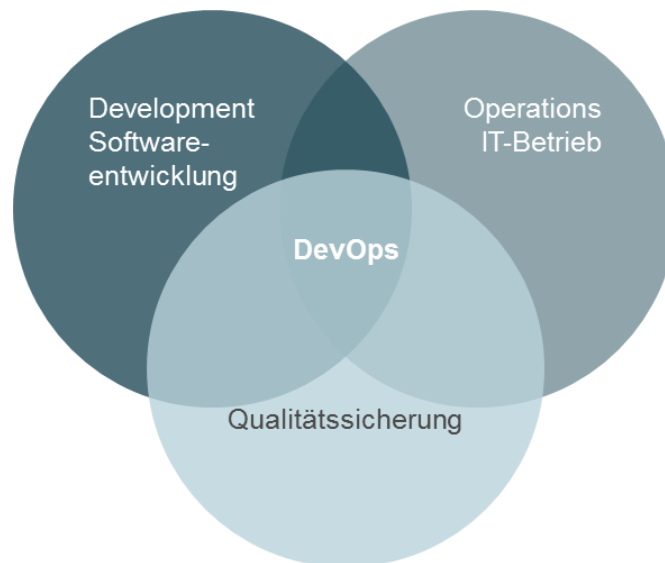
Die Entwickler und Administratoren, die 2009 der Einladung des belgischen IT-Consultant Patrick Debois zum DevOpsDay (<https://www.devopsdays.org/>) nach Gent folgten, kannten dieses Problem.

DevOps, ein Begriff zusammengesetzt aus Development und Operations, also Entwicklung und IT-Betrieb, steht nicht für einen Katalog technischer Maßnahmen, sondern wird von seinen Vertretern vorrangig als Kultur der Silo-Überwindung beschrieben. Es geht darum, in einem partnerschaftlichen Dialog zwischen Entwicklungsabteilung und Admins den IT-Betrieb sowie schnell aufeinanderfolgende Software-Builds abzustimmen. Zentrale Begriffe sind hierbei Continuous Delivery und Continuous Deployment. Ziel ist ein durchgängig abgestimmter Prozess von der Entwicklung bis hin zur Produktivschaltung von Software und deren weitgehender Automatisierung: vom Release-Prozess mit den dazugehörigen Tests bis hin zur Bereitstellung der Ressourcen für einen stabilen Betrieb jeder neuen Version.

Die Etablierung einer DevOps-Kultur erreicht, dass auf der Entwicklerseite die nachgelagerten Prozesse bedacht werden. Hat beispielsweise das Testteam einen anderen Rhythmus, wird die Continuous Delivery ausgebremst. Brauchen die Features eines neuen Builds zusätzliche Datenbanken oder mehr Rechenpower, muss der IT-Betrieb eine Chance haben, diese Ressourcen bereitzustellen. Zugleich bedeutet DevOps die Einführung agiler Methoden im Bereich IT-Operations, damit ein Continuous Deployment möglich wird.

Die DevOps-Kultur erfasst nicht nur die namensgebenden Bereiche Development und Operations, sondern fordert auch frühes Testen, permanentes Monitoring und die ständige Auswertung von Kunden-Feedback sowie eine Planung der Produktentwicklung, die all diese Aspekte unter wirtschaftlichen Gesichtspunkten einordnet. Auch wenn es um Unternehmenskultur, Kommunikation und die Überwindung von Silodenken geht, ist DevOps keineswegs nur ein Konzept, das Reibungsverluste und ungeplante Nacharbeit zur Fehlerkorrektur verringert. DevOps ist die Ausdehnung von Agilität und Lean auf den gesamten Produktlebenszyklus und bietet den

Wettbewerbsvorteil einer verkürzten Time-to-Market bei gleichzeitig hoher Verfügbarkeit und Stabilität sowie sinkenden Betriebskosten.



DevOps bringt verschiedene Verantwortungsbereiche in einen kontinuierlichen Dialog und etabliert einen transparenten Continuous Deployment Workflow.

## 2 DevOps in der Praxis: Maßnahmen in Organisation und Technik

### 2.1 Neues Denken und gemeinsame Verantwortung als Schlüssel

Silo-überwindende Team-Building-Maßnahmen oder gemeinsame Fortbildungen von Entwicklung und Betrieb zu Lean- und agilen Methoden sind hilfreich, um die für DevOps nötige Kommunikation anzutreiben. Und natürlich ist es schön, wenn sich Mitarbeiter beider Teams kennen und verstehen. Aber atmosphärische Maßnahmen allein reichen nicht aus: Das Management muss die organisatorischen und technischen Voraussetzungen schaffen, damit Kommunikation und Prozesse teamübergreifend stimmen.

Zugespielt formuliert: Solange Stabilität das einzige Erfolgskriterium im IT-Betrieb ist, werden die Mitarbeiter sich dort mit aller Macht gegen Störungen schützen – am einfachsten durch die Ablehnung jeder schnellen bzw. nicht penibel vorbereiteten Änderung. Auf der anderen Seite gilt: Wenn einzig die schnelle Fertigstellung von Features den Erfolg der Entwicklung definiert, wird deren praktische Umsetzbarkeit im Betrieb weiterhin wenig Beachtung finden.

Eine der ersten Managementaufgaben bei der Einführung von DevOps ist die Abschaffung von Bewertungskriterien, die Teams gegeneinander ausspielen und abgrenzen. Erst wenn Erfolg so definiert ist, dass er nur durch teamübergreifende Zusammenarbeit zu erreichen ist, kann auch gemeinsam Verantwortung übernommen werden.

### 2.2 Prozessoptimierung

Unabhängig von der technischen Umsetzung der „Delivery Pipeline“ muss eine Organisation den zugrundeliegenden Lean-Gedanken pflegen und Effizienzhindernisse ausräumen. Anzeichen dafür, dass Zusammenarbeit anders organisiert werden muss, sind ungeplante Nacharbeiten<sup>1</sup> oder ein Kommunikations-Overhead. Dies ist der Fall, wenn Tests Fehler an eigentlich fertiger Software aufdecken, die man schon viel früher hätte vermeiden können oder wenn dieselbe Information immer wieder anderen Beteiligten vermittelt werden muss.

In diesem Zusammenhang sei zu erwähnen, dass DevOps-Experten beim IT-Dienstleister keinesfalls eine zusätzliche Kommunikationsebene oder eine Art Verwaltung darstellen dürfen – das würde dem Lean-Gedanken widersprechen. Vielmehr sollte es die Aufgabe sein, direkte Kommunikation zu fördern und Konflikte nachhaltig über Prozessoptimierungen zu schlichten. Die besten DevOps-Experten sind diejenigen, die sich mit der Zeit überflüssig machen.

### 2.3 Technische Hilfsmittel: DevOps-Tool-Kette

In technischer Hinsicht erfordert DevOps vor allem eine Tool-Kette für Continuous Integration, Continuous Delivery und Continuous Deployment auf Basis einer weitgehend virtualisierten IT-Infrastruktur.

---

<sup>1</sup> Die Mitarbeiter von Unternehmen, die maximal ein Software-Release pro Monat veröffentlichen, gaben an, 38 % neue Arbeit zu leisten, 27 % ungeplante und Nacharbeit sowie 35 % andere Arbeiten wie Meetings oder Routinewartung. Bei Mitarbeitern in Unternehmen, die mit mehreren Software-Releases pro Tag mutmaßlich eine ausgeprägte DevOps-Kultur haben, waren es 49 % Neuarbeit, 21 % ungeplante und Nacharbeit sowie 30 % restliche Aufgaben. 2016 State of DevOps Report <https://puppet.com/resources/white-paper/2016-state-of-devops-report>

Agile Softwareteams arbeiten im DevOps-Umfeld beispielsweise mit Jenkins zur kontinuierlichen Integration von Softwarekomponenten. Daran können Build-Automation-Tools wie Apache Ant oder Gradle, sowie Versionsverwaltungssysteme wie Git oder Apache Subversion und Test-Tools wie Emma oder JUnit hervorragend anknüpfen.

Tool Chain und Workflows für die agile, kontinuierliche Bereitstellung der neuen Softwareversionen basieren vor allem auf einem Konzept, das eine weitergehende Automatisierung möglich macht: Infrastructure as Code. Sämtliche benötigten Vorgänge und Konfigurationen zum Aufsetzen von Infrastruktur oder zum Durchführen von Deployments können als Quellcode repräsentiert sowie entsprechend behandelt und entwickelt werden. Damit können die in der Softwareentwicklung üblichen Lösungen zur Qualitätskontrolle auch für den Betrieb eingesetzt werden.

Weiterer Vorteil des Infrastructure as Code-Ansatzes ist die zentrale Verwaltung des Quellcodes unter Einsatz einer Versionskontrolle. Darüber hinaus kann die Konfiguration von Servern sowie virtuellen Maschinen automatisiert getestet und so die Qualität wie auch die Wiederholbarkeit eines Deployments sichergestellt werden. Eine beliebte Tool-Kombination ist das Systemmanagement-, Provisioning- und Monitoring-Tool Foreman zusammen mit Konfigurationsmanagement-Tools wie Puppet, Chef oder Salt.

Sind die IT-Infrastruktur und Umgebungen für Entwicklung, Test und Betrieb komplett (oder weitestgehend) virtualisiert, können die für ein neues Build ggf. erforderlichen zusätzlichen Ressourcen oder Neukonfigurationen komplett über APIs oder entsprechende Konsolen durchgängig automatisiert und ohne manuelle Eingriffe in der Infrastruktur bereitgestellt werden.

## 3 DevOps bei einem Outsourcing des IT-Betriebs an IT-Dienstleister

### 3.1 Ausgangspunkt

Der bloße Umstand, dass es eine erfolgreiche DevOps-Bewegung gibt, zeigt bereits, dass eine optimierte Zusammenarbeit zwischen Entwicklung und IT-Betrieb bereits innerhalb einer Unternehmung eine kulturelle Herausforderung darstellt. Sind Gesamtverantwortung, Entwicklung und Betrieb durch Outsourcing-Partnerschaft auf zwei (z. B. entwickelnder Anbieter und IT-Dienstleister) oder sogar auf drei Unternehmen (z. B. Webshop-Betreiber, Webshop-Entwickler und IT-Dienstleister) verteilt, wächst die Komplexität bei den Schnittstellen sowie bezüglich Organisation, Methoden, Tools und Prozessen weiter.

Outsourcing ist aber bei vielen Betreibern von Cloud-Anwendungen und Webshops elementarer Bestandteil des Geschäftsmodells. Die Anbieter können sich so auf ihr Kerngeschäft fokussieren. Die Outsourcing-Partner sichern die Skalierbarkeit und Zuverlässigkeit der IT-Infrastruktur auf Basis von kalkulierbaren Kosten – und befreien das Unternehmen damit von Investitions- und Betriebsrisiken. Viele Unternehmen haben den IT-Betrieb komplett ausgelagert und fahren kommerziell wie technisch gut damit, das Management von Servern, Storage, Netzwerkinfrastruktur, Connectivity, Sicherheits- und Betriebssystemen in die Hände von Betriebsspezialisten zu legen. Tendenziell aber erhöht das Outsourcing zunächst die Mauer zwischen IT-Betrieb und Softwareentwicklung.

Wie kann ein Unternehmen die Vorteile einer agilen Entwicklung mit einer ausgelagerten IT-Infrastruktur bei einem IT-Dienstleister nutzen?

### 3.2 DevOps als Dienstleistung von Premium-IT-Dienstleistern

Erste, besonders innovative IT-Dienstleister reagieren jetzt auf die wachsenden Anforderungen aus den Entwicklungsabteilungen ihrer Kunden. Sie etablieren eigene DevOps-Experten im Bereich IT-Infrastruktur, die die Zusammenarbeit mit den Entwicklungsteams des Kunden deutlich intensivieren, den Release-Prozess in Form einer Deployment Pipeline automatisieren und optimieren. Vorreiter in Deutschland ist hier die noris network AG.<sup>2</sup>

In der Praxis bedeutet das auf der technischen Seite, dass der Dienstleister Deployment Pipelines aufbaut, die vom Unit-Test über Integration-Tests bis zum automatisierten Deployment in der Produktion reichen. Die Schnittstelle zwischen Entwicklung und Betrieb wird dabei durch die Deployment Pipeline sowohl explizit definiert als auch überbrückt. Die dadurch mögliche weitgehende Automatisierung des Release-Prozesses macht diesen nicht nur in hoher Frequenz wiederholbar, sondern minimiert zugleich Fehlerrisiken im Betrieb.

DevOps in einem Dienstleistungsverhältnis bedeutet eine ungewöhnlich enge Zusammenarbeit. Aufseiten des Outsourcing-Partners müssen erfahrene DevOps-Spezialisten bereitstehen, die gemeinsam mit dem Kunden ein Team bilden. Es handelt sich dabei um Mitarbeiter, die ein tief reichendes Verständnis für Software und die Prozesse seiner Entwicklung haben. Aus diesem Verständnis heraus können sie die Anforderungen des stabilen Betriebs in die Applikationsentwicklung einbringen. Wichtig: Die Arbeit der DevOps beginnt idealerweise bereits zu einem sehr frühen Zeitpunkt – auch wenn hier typischerweise der Umfang des geplanten Hostings noch nicht einmal genau bekannt ist.

---

<sup>2</sup> Berichte hierzu unter anderem in der Funkschau 2015/01-08 und IT-DIRECTOR 2016/01-02



Wie wird der Softwareentwickler die Software designen und implementieren und wie soll das Gesamtsystem nach der Projektphase gepflegt werden? Kann der Softwarespezialist die Anforderungen an Hardware und Konnektivität richtig einschätzen? Gibt es besondere Sicherheitsanforderungen? Die DevOps-Spezialisten evaluieren diese Themen und fachlichen Anforderungen gemeinsam mit den Product Ownern, den Softwareentwicklern und ggf. mit Spezialisten aus dem IT-Team.

Je früher diese Einbindung und Diskussionen stattfinden, desto besser: Für die Etablierung der DevOps-Kultur müssen alle Abteilungen von Beginn an zusammenarbeiten. Ein Grund: Gerade in der frühen Designphase der Softwareentwicklung fallen Entscheidungen mit erheblichen Auswirkungen auf Testprozesse und den späteren Betrieb. Ein weiterer Grund: Je früher die Tool-Ketten abgestimmt werden, desto leichter fällt der Übergang in den produktiven Betrieb der Software.

Mit modernen Virtualisierungs- und Containertechnologien, mit skalierbaren „Cloud“-Lösungen können Festlegungen zum Ressourcenbedarf vielfach auf später verschoben werden, aber die Grundlagen für eine spätere Skalierbarkeit des Systems müssen frühzeitig gelegt werden. Bei zeitkritischen Projekten, für erste Tests und User Feedbacks kann man mit einem Minimum Viable Product beginnen. Dieses ist auf Kernfunktionalitäten beschränkt und weitere Funktionen werden in möglichst kleinen Inkrementen und kurzen Iterationen nachgeliefert – mit einer frühzeitig abgestimmten Continuous-Delivery-Strategie.

## 4 Praxiserfahrungen: Kostenvergleich klassischer Betrieb versus DevOps

Die bisherigen Ausführungen machen deutlich: Wer die Wettbewerbsvorteile einer agilen Softwareentwicklung zuverlässig und schnell in den Betrieb übernehmen und damit seinen Kunden bereitstellen will, findet dafür in DevOps-Experten und -Dienstleistungen seines IT-Dienstleisters wirksame Unterstützung. Aber wie steht es um die Kosten? Wie schlagen Teams, an denen Mitarbeiter des Dienstleisters beteiligt sind, und technische Einrichtungen für das Continuous Delivery und Continuous Deployment wirtschaftlich zu Buche?

Um hierzu valide Aussagen zu bekommen, wurden Investitionskosten und Betreuungsaufwände von vergleichbaren Setups zweier Kunden der noris network gegenübergestellt. Beide Applikationssysteme bestehen aus ca. 500 virtuellen Servern und fünf bis sieben unterschiedlichen im Verbund betriebenen Applikationen.

Klassischer Betrieb einer geschäftskritischen Applikation	Betrieb als DevOps-Kooperation
<p>Aufbau des Setups in physisch getrennten Umgebungen für Entwicklung, Qualitätssicherung- und Produktion.</p> <ul style="list-style-type: none"> <li>• Hardware- und Softwareausstattung muss entsprechend ausgelegt werden</li> <li>• Einsatz von 50 bis 60 Hypervisoren</li> <li>• Einsatz 20 dedizierter Security-Hardwaresysteme</li> <li>• 8-10 dedizierte Datenbankserver (Einsatz dedizierter Datenbanken auf dedizierten Hardwaresystemen aus kommerziellen lizenzrechtlichen Gründen)</li> <li>• 40 bis 60 Core-Lizenzen von Oracle Enterprise</li> <li>• Nicht nach dynamischen Auslastungen skalierbar</li> <li>• Wenige dynamische Skalierungsmöglichkeiten, starr im Sinne der Technik und der kommerziellen Möglichkeiten</li> </ul>	<p>Aufbau des Setups in physisch getrennten Umgebungen für Entwicklung, Qualitätssicherung und Produktion. Durch die Möglichkeit der automatischen Bare-Metal-Konfiguration können Hardwareelemente allerdings wesentlich einfacher und schneller in andere Umgebungen verschaltet werden. Die Softwarebetankung kann dann anschließend sofort durchgeführt werden. Durch den Einsatz von nicht extrem limitierten Datenbanksoftwareprodukten kann eine einheitliche Hypervisorstruktur aufgebaut werden. Durch die Konsolidierung und Virtualisierung vieler Hardware-Komponenten kann die Anzahl reduziert werden.</p> <ul style="list-style-type: none"> <li>• Anzahl der Hypervisoren kann geringer ausgelegt werden</li> <li>• Einsatz 10 dedizierter Security-Hardwaresysteme</li> <li>• Skalierungsmöglichkeiten durch die Verschaltungsart der Hypervisoren</li> <li>• Skalierung der Datenbanken durch die gleichen Hypervisoren und offenen Lizenzmodelle der Datenbankhersteller</li> <li>• Dadurch insgesamt etwas geringerer Hardwareaufwand</li> </ul>
<b>Initiale Einrichtung:</b>	
<ul style="list-style-type: none"> <li>• Feststehendes einfaches Setup, einfach integrierbare Halbautomatisierung</li> </ul>	<p>Für die Festlegung der Architektur des Systems muss mit erhöhten planerischen Aufwendungen gerechnet werden. Die Bewertung des Setups</p>

<ul style="list-style-type: none"> <li>• Relativ schnelle Installation und Migration und damit 3-4 Monate Vorlauf</li> </ul>	<p>und dessen Aufbau sind deutlich aufwendiger und müssen mit fachlich kompetenten Spezialisten begleitet werden. Das Setup muss den agilen Anforderungen entsprechen, um die Mehrwerte für die Entwicklung wie auch den Betrieb abbilden zu können.</p>
<b>Betriebsaufwendungen:</b>	
<p>Starres, streng dokumentiertes Betriebsverfahren. Speziell bei Änderungen (Change Requests) werden sowohl an den Auftraggeber wie den Provider hohe Anforderungen gestellt, um keine Incidents (Ausfälle) oder Security Incidents zu erzeugen. Die fachliche wie auch organisatorische Kompetenz muss auf beiden Seiten hoch sein. Gleichzeitig muss die Verlässlichkeit wie auch die pedantische kontinuierliche Abarbeitungsfähigkeit vorhanden sein. Die Change Requests müssen über mehrere Hierarchien geprüft und genehmigt werden.</p> <ul style="list-style-type: none"> <li>• Gefahr: Personal entweder überfordert oder unterfordert</li> <li>• Aus diesem Grund zusätzlicher Aufwand im 4-Augen-Prinzip</li> <li>• Erhöhte Kontrolle über den Genehmigungsprozess</li> <li>• Erhöhter Aufwand im Change Advisory Board (CAB)</li> <li>• Wegen des relativ hohen Risikos werden die CRs meistens in den weniger betriebskritischen Zeiten durchgeführt</li> <li>• Daraus resultierend sind die Wartungsfenster völlig ausgebucht und ein CR-Stau findet statt</li> <li>• CRs innerhalb der Betriebszeiten sind immer als Risiko zu bewerten</li> <li>• Auslastung des Fachpersonals von beiden Seiten</li> <li>• Erhöhte Personalkontrolle von beiden Seiten, um keine gefährliche Überlastung des Personals zu erzeugen</li> </ul>	<p>Durch den vorangegangenen durchgängigen Aufbau einer Continuous Deployment Pipeline sind Standard-Changes einfach durchzuführen. Die Qualitätsprüfungen finden während der Entwicklung statt. Ein Großteil der Changes kann im normalen Administrationsfenster durchgeführt werden.</p> <ul style="list-style-type: none"> <li>• Deutlich geringeres Change-Risiko</li> <li>• Anzahl der Changes kann deutlich erhöht werden</li> <li>• Deutliche Reduzierung der Ausführungsaufwendungen</li> <li>• Deutliche Reduzierung der administrativen Überwachung</li> </ul>
<b>Aufwandsanalyse:</b>	
<p>Kosten der Hypervisoren initial mit Software: 1.250.000 EUR</p>	<p>Kosten der Hypervisoren initial mit Software: 500.000 EUR</p>
<p>Kosten der Datenbank-Lizenzen nach Listenpreis Oracle: 2.000.000 EUR</p>	<p>Keine</p>
<p>Wartungskosten der Datenbank nach Oracle Listenpreis: 400.000 EUR pro Jahr</p>	<p>Wartungskosten der Datenbank sind reine Betriebsführungskosten und erfordern keine Lizenzkosten.</p>

Initialer Personalaufwand:	
<b>Aufbau und Migration:</b> 2000 Stunden	<b>Aufbau und Implementierung der Continuous Deployment Pipeline:</b> 10 000 Stunden
Betriebsführung:	
200 Stunden pro Monat	300 Stunden pro Monat (100 Stunden Mehraufwand für die Wartung der Continuous Deployment Pipeline)
Change-Request-Aufwand:	
50 bis 80 Changes pro Woche Aufwand pro Change 0,5 bis 2 Stunden im 4-Augen-Prinzip	Beliebig viele Changes werden im Continuous Deployment eingespielt
<b>Aufwand für Provider pro Monat:</b> 600 bis 800 Stunden	<b>Aufwand für Provider pro Monat:</b> 100 Stunden durch teilweise nötige Begleitung des Deployments und der automatisierten Anpassung der Infrastruktur
<b>Bezifferter Aufwand auf Provider-Seite:</b> 1000 Stunden pro Monat	<b>Bezifferter Aufwand auf Provider-Seite:</b> 400 Stunden pro Monat
<b>Ausgehend von diesen 600 Stunden Differenz in den Personalkosten ist der Return on Investment (ROI) für die initial deutlich personalaufwendigere DevOp-Lösung nach knapp über einem Jahr erreicht (aus Sicht der Kosten auf Seiten des Providers).</b>	

Der Vergleich zeigt deutlich: Das initiale Investment – Zeit und Kosten – ist im DevOps-Modell und durch den Aufbau der erforderlichen Continuous Delivery/Continuous Deployment Pipeline deutlich höher. Diese Kosten wären ggf. nochmals höher, wenn der Kunde all diese Maßnahmen und Infrastrukturen in Eigenregie hätte realisieren müssen. Also nicht wie bei diesem Setup von der langjährigen Erfahrung der noris network mit ihren DevOps-Experten, dem technischen Know-how bei Tool-Ketten sowie dem technischen Stand der Virtualisierungsplattformen profitiert hätte.

Ganz anders stellt es sich dann bei den laufenden Kosten im Betrieb dar. Werden über agile Entwicklungsmethoden Builds in hoher Zahl in den Betrieb übernommen, so rentiert sich der Aufbau der Pipeline schnell, weil deren „manuelle“ Übernahme im klassischen Modell enorm viel Personaleinsatz beim Outsourcing-Partner und damit hohe Kosten verursacht.

Bei den hier angenommenen 50 bis 80 Builds/Patches wird der monatliche Aufwand im Betrieb mehr als halbiert. Bei einer noch höheren Anzahl an Builds wird dieser Einsparungseffekt noch größer, weil die Grenzkosten für das Einspielen dieser zusätzlichen Builds praktisch auf Null sinken – während der manuelle Aufwand im klassischen Modell praktisch gleich hoch bleibt.

Fazit: Nach etwas über einem Jahr Betrieb hätte die DevOps-Installation die initialen Mehrkosten durch die niedrigeren Betriebskosten wieder „eingespielt“. Dabei sind hier die Pflegeaufwendungen für die Continuous Delivery/Continuous Deployment Pipeline mit 100 Personenstunden/Monat eher konservativ hoch angesetzt.

## 5 Eignung für DevOps

Ein klassischer Betrieb bleibt somit dann sinnvoller, wenn die Anforderungen der Anwendung im Wesentlichen mit einem einmaligen Aufbau und mit wenigen Änderungen in der Software sowie beim Ressourcenbedarf/IT-Infrastruktur erfüllt werden können. In diesen Fällen wird sich ein agiler IT-Betrieb und der Aufbau einer Pipeline nicht oder nur sehr langfristig rechnen. Beispiele wären etablierte ERP-Systeme oder Backofficesysteme wie Sharepoint und Mail.

Hat eine Software hingegen eine hohe Änderungsfrequenz, weil sie beispielsweise auf volatile Märkte wie Webshops oder Cloud-Anwendungen zielt, sind agile Entwicklung und der darauf getaktete Betrieb fast schon zwingend. Gut geeignet ist der Ansatz auch bei Green-Field-Entwicklungen und anfänglich unbekanntem Skalierungsanforderungen. Stetiges Wachstum, das Elastizität bei den Ressourcen verlangt, wird ebenfalls durch DevOps und Infrastructure as Code besser unterstützt als durch klassische starre Betriebsformen. Besonders Software-Start-ups sollten sich einen DevOps-erfahrenen IT-Dienstleister suchen.

## 6 Fazit

Die Anforderungen an die Sicherheit, Verfügbarkeit und Skalierbarkeit ihrer IT-Infrastruktur sind heute für viele Unternehmen so hoch, dass es ohne Outsourcing und die Unterstützung durch IT-Dienstleister kaum noch geht – sowohl aus technischer wie auch aus wirtschaftlicher Sicht. Wenn IT-Dienstleister DevOps-Spezialisten haben, die mit den Entwicklern des Kunden (extern oder intern) ein Team bilden, so ist das Outsourcing auch für Unternehmen, die agile Software entwickeln, möglich, ja sogar empfehlenswert.<sup>3</sup>

Mit der Bereitstellung erfahrener DevOps-Spezialisten durch den IT-Dienstleister ist die Einrichtung einer Continuous Delivery/Continuous Deployment Pipeline tendenziell schneller und zuverlässiger realisierbar. Der initiale Investitionsbedarf für die Etablierung einer solchen DevOps-Kultur, für die Prozessoptimierung und den technischen Aufbau der angesprochenen Pipeline, bleibt aber deutlich höher als bei klassischen Installationen mit ihren Change-Request-Prozessen. Diese höheren Anfangsinvestitionen amortisieren sich aber rasch – abhängig von der Größe der Installation und der Zahl der agil entwickelten und in Betrieb übernommenen Software-Builds. Der ROI dürfte bei Beteiligung von DevOps-erfahrenen Mitarbeitern des IT-Dienstleister, die ihr Operating-Praxiswissen in den querschneidenden Entwicklerteams des Kunden einbringen, typischerweise nach ein bis zwei Jahren erreicht sein.

---

<sup>3</sup>Leistungsstarke IT-Organisationen, die 30-mal häufiger Releases ausliefern als andere, tun dies mit 200-mal kürzeren Durchlaufzeiten. Dabei ist das Ergebnis ihrer Arbeit auch noch besser: Sie erleben 60-mal weniger Ausfälle und erholen sich von Fehlern 168-mal schneller. 2015 State of DevOps Report  
<https://puppet.com/resources/white-paper/2015-state-devops-report>

## Die noris network AG

1993 gegründet zählt die noris network AG zu den deutschen Pionieren auf dem Gebiet moderner IT-Dienstleistungen und bietet Unternehmen maßgeschneiderte ITK-(Informations- und Telekommunikationstechnik-)Lösungen in den Bereichen IT-Outsourcing, Managed Services, Cloud Services sowie Colocation. Technologische Basis dieser Services ist eine leistungsfähige IT-Infrastruktur mit einem noris network-eigenen Hochleistungs-Backbone und mehreren Hochsicherheitsrechenzentren – darunter das Rechenzentrum NBG6, eines der anerkannt modernsten und energieeffizientesten Rechenzentren Europas. Unter der Marke datacenter.de werden ergänzend zu den Individuallösungen standardisierte Premium-Rechenzentrumsprodukte angeboten. Mitte 2017 wurde außerdem das hochmoderne Rechenzentrum MUC5 in München Aschheim eröffnet. Das Hochsicherheitsrechenzentrum wurde bereits mit dem Deutschen Rechenzentrumspreis 2016 in der Kategorie "Neu gebaute energie- und ressourceneffiziente Rechenzentren" ausgezeichnet.

Die noris network AG ist mit ihrem integrierten Managementsystem nach ISO/IEC 27001 (Informationssicherheit), nach ISO/IEC 20000-1 (Servicemanagement) und nach ISO 9001 (Qualitätsmanagement) für alle Geschäftstätigkeiten zertifiziert. Außerdem haben die noris network-eigenen Rechenzentren das ISO 27001-Zertifikat auf Basis von IT-Grundschutz des BSI (Bundesamt für Sicherheit in der Informationstechnik) erhalten. Die hochverfügbaren Rechenzentren der noris network tragen die maximal mögliche Anzahl von fünf Sternen, die der eco (Verband der deutschen Internetwirtschaft e. V.) im eco Datacenter Star Audit zu vergeben hat.

Die herausragenden Lösungen und Services der noris network eignen sich im ganz besonderen Maße für Unternehmen aus den Sektoren E-Commerce und Finance sowie Unternehmen, die hohe und höchste Anforderungen an Sicherheit und Hochverfügbarkeit stellen.

Typische Kunden der noris benötigen höchste Datensicherheit, zuverlässigen Betrieb und kompetente, zertifizierte Dienstleistungen rund um die Uhr, um die vitalen Kernsysteme ihres Geschäftes im Betrieb zu halten.

### **noris network AG**

Thomas-Mann-Straße 16-20

90471 Nürnberg

Tel.: +49 911 9352 - 0

**noris network**

[www.noris.de](http://www.noris.de)