

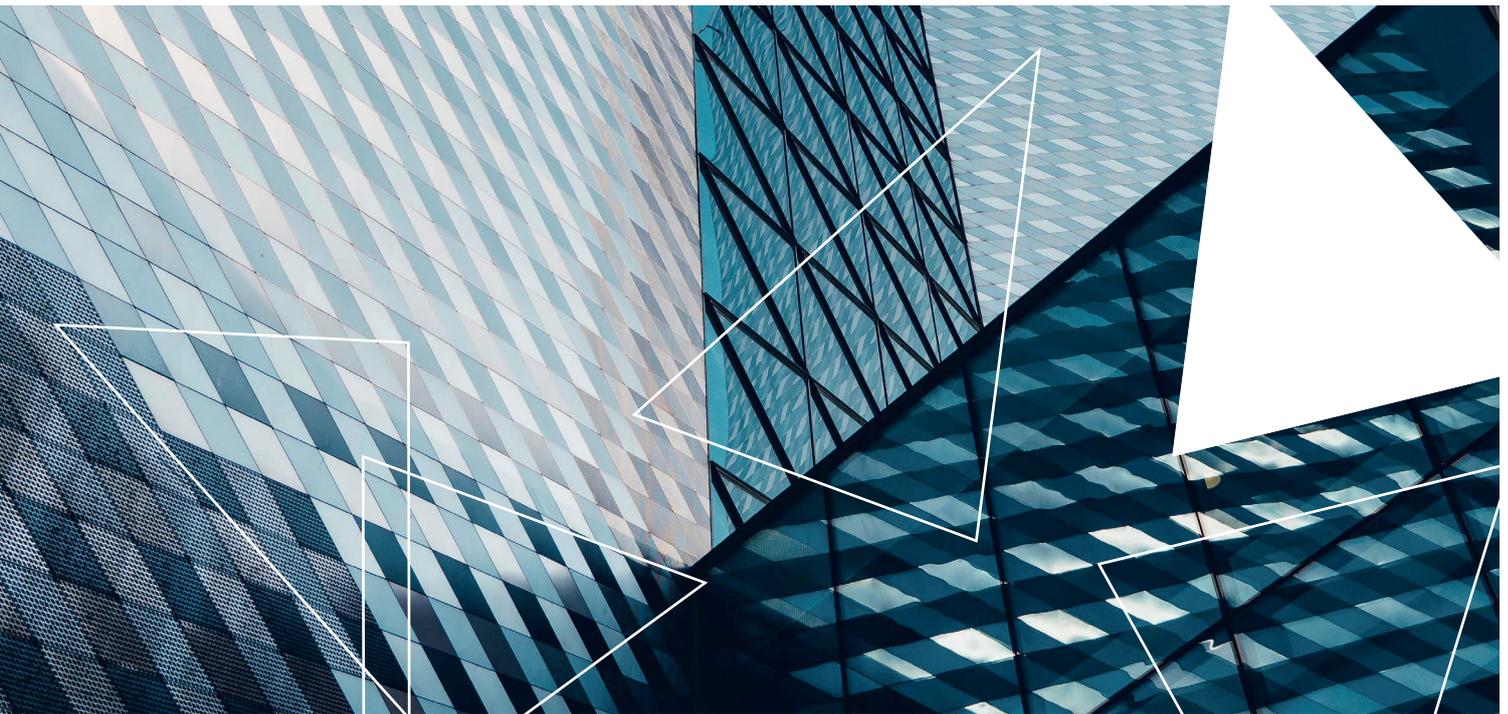
# PROFI.info

**DEVOPS**

EIN SINNVOLLES BETRIEBSSYSTEM  
DER MODERNEN IT?

# INHALT

<b>DevOps – Eine Frage von Kultur und Organisation</b>	<b>3</b>
<b>Entwicklung</b>	<b>5</b>
<b>Betrieb</b>	<b>6</b>
<b>Gemeinsames Denken und Arbeiten</b>	<b>8</b>
<b>Empfehlung</b>	<b>10</b>
<b>Zusammenfassung</b>	<b>12</b>



# DEVOPS – EINE FRAGE VON KULTUR UND ORGANISATION

Über den Wert, den DevOps Unternehmen bringen kann, scheint Einvernehmlichkeit zu herrschen. Keine Studie, keine Referenzstory, die nicht von effizienterer und effektiverer Entwicklung und Bereitstellung von Software an den Endanwender spricht. In Zeiten von Cloud sowieso. Ein Blick in die Details vieler DevOps „Projekte“ offenbart jedoch, dass die meisten Firmen und deren IT-Organisationen aktuell eher bei Dev & Ops stehen – also einer „engeren“ Zusammenarbeit zwischen Entwicklung und Betrieb, als dass sie tatsächlich schon DevOps – einschließlich Infrastrukturdefinition als zentralen Bestandteil des Entwicklungsprozesses (von der „Beta“ über den „Release-Candidate“ bis zum ausgelieferten Produkt) aus einem Guss umsetzen.

DevOps ist die Beschreibung eines auf vollständige Automatisierung ausgerichteten Denkens und Handelns, alle Zeilen Code, die eine Funktion – einen Nutzen – darstellen, sofort zu testen und verfügbar zu machen um damit Mehrwert zu stiften. DevOps sieht keine Abteilung Entwicklung und keine Abteilung Betrieb vor. Doch die sind in den meisten Unternehmen nun mal da und für all die Anwendungen konzipiert, die von Dritten fertig getestet zur Installation, Betrieb und Nutzung vorgesehen sind. Wir sprechen von Standardanwendungen wie SAP, Microsoft oder branchenspezifischen Anwendungen vieler Hersteller. Im Rahmen des Standardisierungsgedankens wird auch individuell entwickelte Software der Entwicklungsabteilungen ebenfalls fertig getestet

bereitgestellt. Weil dabei immer noch oft nach Pflichten- und Lastenheft vorgegangen wird, sind die Entwicklungszyklen mit einhergehendem Rollout eher überschaubar.

Da nun im Rahmen der „Digitalisierung“ Unternehmen vermehrt eigene Software, Apps, Microservices, APIs und so weiter entwickeln, entsteht eine ganz andere Anforderung: Entwickelte Software muss nun auch in vielen Anwenderszenarien getestet und dann bereitgestellt werden.

**„Digitalisierung“ führt auch dazu, vormals analoge Produkte durch das Anreichern mit softwarebasierten Komponenten zu digitalen Produkten zu entwickeln. Diese Services – ob nun als App, als API, als „embedded function“, als Feature einer Sprachsteuerung oder eines Smart-TVs – gilt es ständig weiter zu entwickeln und Anpassungen an den unterstützten Plattformen sehr zeitnah zu integrieren.**

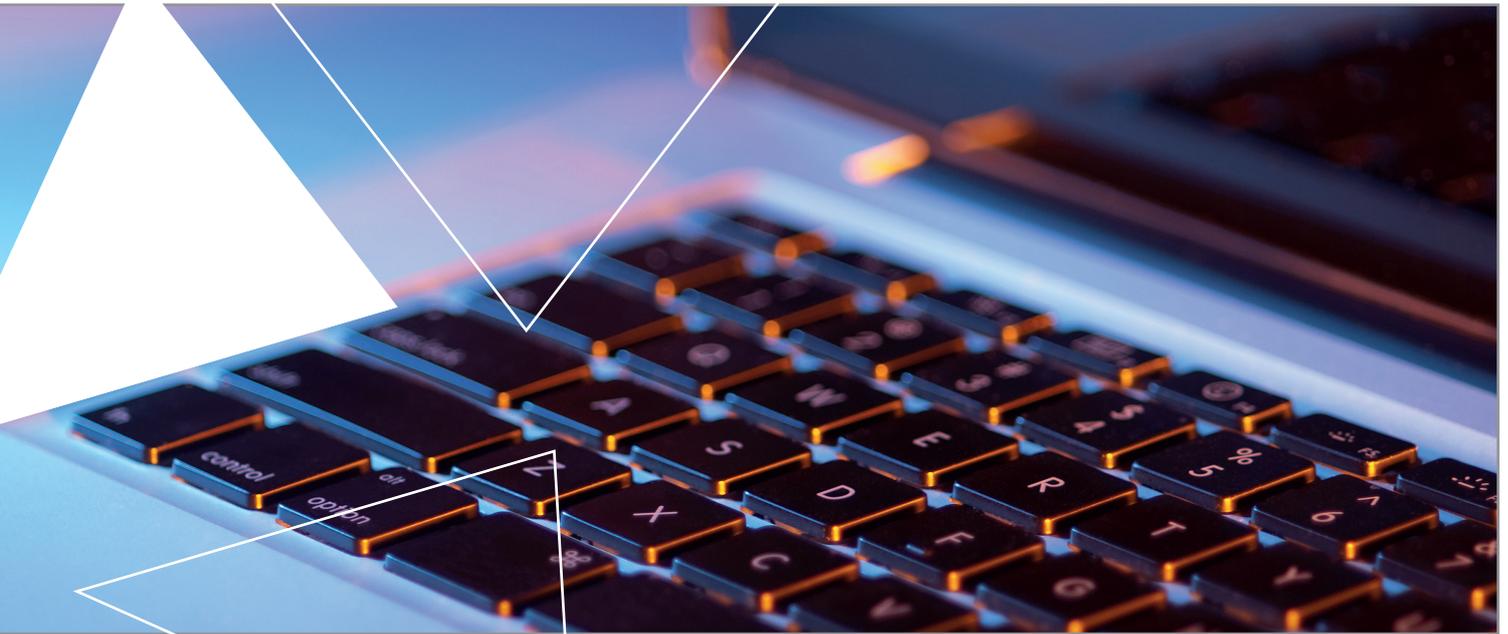


Weil digital auch agil sein soll, passiert dies in kleinen Stücken innerhalb kurzer Intervalle (Inkremente und Sprints). Hier kristallisiert sich bereits die erste Kulturfrage heraus: „Wie kann man nur so fehlerhafte Software entwickeln“ versus „wie kann man nur so divergente Test- und Live-Systeme haben“? Dies sind die extremen Reaktionen der Seiten Dev und Ops. Die propagierte Digitalisierung hat jedoch Priorität und so nähert man sich an. Oder doch nicht? Anbieter von cloud-basierten Plattformen und Services erleichtern der Dev-Seite den Zugriff auf Ressourcen ohne die als hinder-

lich empfundenen Regeln des internen Betriebes beachten zu müssen. Der Zoo an benutzten Services, die durch diese Vermeidungsstrategie interner Regeln und Prozesse entsteht und deren Kosten, machen die Nutzung von Cloud-Anbietern deswegen aus Gesamtsicht aber noch lange nicht effizient. Andererseits wird im Betrieb antizipiert, dass die „Funktionen“ der Cloud attraktiv für die Entwickler sind und die bestehende interne Infrastruktur wird dann „cloudifiziert“, indem Standardsoftware/-plattformen dafür implementiert werden. Günstiger wird das in der Regel

auch nicht, denn es gibt ja nun einen zusätzlichen Service, der abgerechnet werden muss.

Werden damit die grundlegenden Fragen oder besser die effizienzsteigernden Ideen von DevOps real? Mitnichten! Denn sowenig wie „Cloud“ ein Ort ist, ist DevOps eine Art Tool. Wenn man sich ein Miteinander aus Dev und Ops zum Ziel setzt, muss man die jeweiligen Interessen und unterschiedlichen Ziele verstehen und am besten mit Synergie zusammenbringen.



## ENTWICKLUNG

Im Kern sind die Anforderungen aus Entwicklungssicht das Entwickeln, Testen, Releases und Updaten von Software(komponenten) unter Sicherstellung des Qualitätsanspruchs an Software, um in enger Kooperation mit oder in den Fachbereichen so die Digitalisierung von Produkten und Prozessen Realität werden zu lassen. Dabei sollen komplette Testumgebungen sowohl aus Hardware (z. B. Mobiltelefone, Tablets, Notebooks/PCs, Server) aber besonders auch deren Clients (z. B. Browser, Betriebssysteme für mobile und stationäre Geräte) für den Tester einfach zur Verfügung gestellt werden. Auch die Evaluierung, das Ausprobieren von neuen technischen Möglichkeiten (Rapid Prototyping) soll ermöglicht werden, z. B. durch einfaches Bereitstellen einer neuen Funktion oder Anwendung oder deren Services dafür. Entwickler wollen und müssen Zugriff auf Teile der DevOps-Landschaft wie z. B. Monitoring, Logging etc. Zugriff haben, um selbstständig Verbesserungen am Code vornehmen zu können.

Fehlende produktionsgleiche Testumgebung und Integrationsumgebung führen innerhalb der Entwicklung dazu, dass das tatsächliche Verhalten einer Anwendung oft erst im Produktionsbetrieb erkannt wird und dann eine Menge Incidents im Betrieb erzeugt. Schnelle Reaktionen auf diese Fehler sind jedoch durch langsame, niederfrequente Deployment-Möglichkeiten nicht so ohne Weiteres möglich. Die Pflege und Wartung der CI/CD-Pipeline Umgebung, die ja jetzt quasi die Infrastruktur der Entwicklungsteams darstellt und natürlich auch als solche betrieben werden muss, sehen Entwickler nicht als Teil ihres Jobs an. Die an sich gewünschte Standardisierung und Aktualisierung dieser Umgebung wird stiefmütterlich behandelt und die daher notwendige Handarbeit erhöht den Betriebs-/Entwicklungsaufwand on top der eigentlichen Entwicklung.

Der Nutzen von DevOps für Entwicklungsteams liegt nicht nur darin, schnell die größtmögliche, komplette, produktionsnahe Testumgebung zur Qualitätssicherung einfach verfügbar zu haben, sondern auch eine skalierbare CI/CD Pipeline – je mehr Projekte, je mehr muss die CI/CD Pipeline mitwachsen – aus Coderepositories, Artefakterepository, CI/CD zu haben, deren Umgebung nicht von Entwicklern selbst betreut werden muss. Das erlaubt eine höhere Frequenz von Deployments zu geringen Kosten.

Agile Teams können durch Fokussierung auf das Entwickeln die Sprintergebnisse verbessern und dabei die „Definition of Done (DoD)“ einfacher erreichen. Einfaches, schnelles Ausprobieren trägt zudem sowohl zur Optimierung der Anwendung und des Business Case in Zusammenarbeit mit der Fachabteilung bei, weil MVPs (Minimal Viable Product) besser validiert werden können.



## BETRIEB

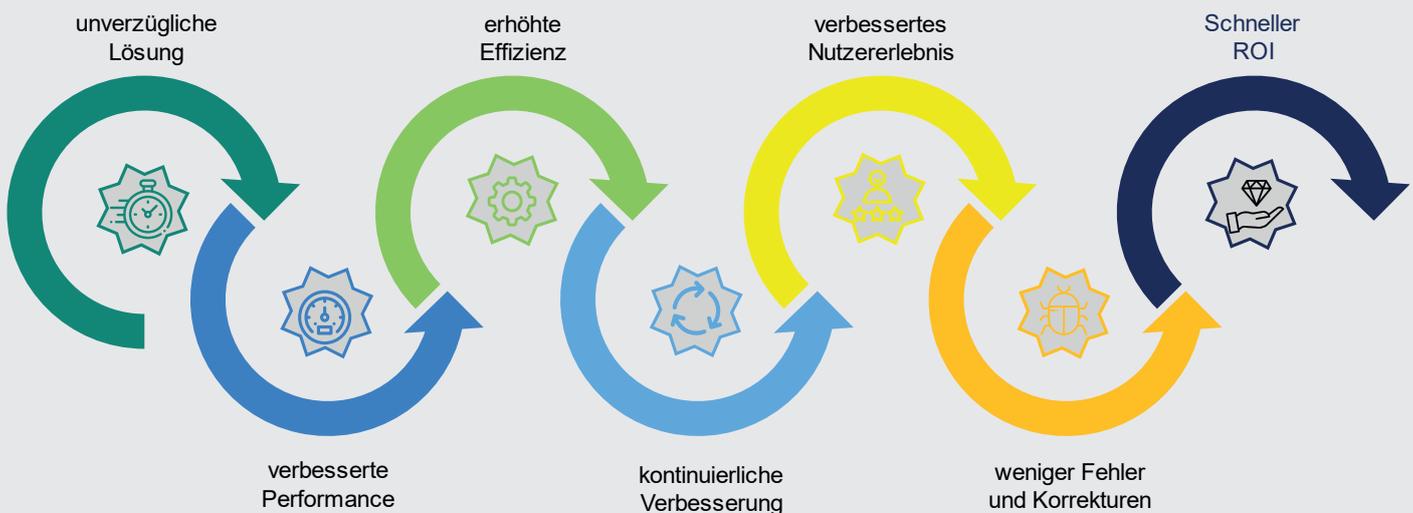
Im Betrieb will man die Infrastruktur für die Entwicklung und den stabilen und sicheren Betrieb der Anwendungen sicherstellen. Dazu gehört, skalierbare Infrastrukturen bereitzustellen, deren Lifecycle zu managen, die Kompatibilität der Systeme sicherzustellen und das Ganze unter der Vorgabe gegebene Budgets einzuhalten. Daher stoßen im Betrieb neue Anforderungen und Projekte gerne erst mal auf Widerstand, zumal die ungeplanten Arbeiten durch Changes und Incidents mit mehr Software-Projekten steigen und durch Einbezug des Kunden/Anwenders in agilen Initiativen und Innovationsmethoden die Volatilität in den Anforderungen zusätzlich gemanagt werden müssen. Gegenüber dem klassischen Planungspro-

zess fehlt zudem zunächst auch oft die sonst klar eingesteuerte Auftragslage aus den Fachabteilungen. Eine klare, transparente Zuteilung und Verrechnung von Betriebskosten zu diesen Projekten ist in den meisten Fällen nicht möglich.

Das Feedback oder Eingeständnis vieler Betriebsverantwortlicher ist auch, dass durch die klassische Art den Betrieb zu organisieren, geeignete Skills für DevOps und dessen Kultur, Methoden und Tools im Betrieb nicht ausreichend vorhanden sind, um der Dynamik der DevOps-Entwicklung standzuhalten. Frameworks wie ITIL (bis V3), bestehende Governance-Vorgaben und Prozessdefinitionen lassen zudem wenig Spielraum für „andere“

Prozesse. Der Versuch, die „Welt der 2 Geschwindigkeiten“ (Bimodale IT) in einem einzigen „Betriebsmodell“ auf organisatorischer Ebene zu bedienen, scheidet zudem zunehmend. Am Zustand wenig standardisierter Managementumgebungen für Testing, Staging und Livebetrieb der gesamten IT-Landschaft, die sowohl den existierenden, aber auch den zusätzlichen neuen Anwendungen bereitgestellt werden müssen, ändert sich ohne neue Vorgehensweisen aber nichts. Im Gegenteil: Die Betriebsaufwände, die durch manuelles Be- und Abarbeiten der Incidents und Changes schon nicht niedrig sind, nehmen mit dem Mehr an Anwendungen im Betrieb zu.

Mit DevOps stellen Unternehmen den Kunden mit seinen Erwartungen bei digitaler Produktentwicklung in den Vordergrund. Schnelles Lernen braucht schnelles Liefern und andersrum. Kunden erleben dabei den Mehrwert in einer an ihren Wünschen und Feedbacks ausgerichteten Entwicklung direkter und sind dann auch gegenüber vermeintlichen Fehlern oder Feature-Lücken positiver eingestellt, wenn sie erleben, dass „sich etwas bewegt“.



Aus guter Erfahrung heraus, versucht man für den Betrieb dann eine „DevOps-Standard-Plattform“ zu etablieren, deren Kosten (i. d. R. Lizenzen) zunächst belasten und deren Einführung nach Standardvorgehen auch einige Zeit und Ressourcen in Anspruch nimmt, ehe ein positiver Effekt in den Betriebskosten entsteht. Und da Betrieb nicht als Innovations-Enabler gesehen wird, gibt es für die notwendigen „internen Innovationen“ kaum Budgets – kurzgefasst: Nur wegen DevOps gibt es kein Budget.

Aber DevOps trägt dazu bei, die notwendige Standardisierung für hybride Infrastrukturen mit hybriden Cloud-Modellen zu schaffen, dabei die dafür nötigen Betriebsaufwände zu senken, weil man die Automatisierung und damit auch die Skalierbarkeit erhöhen kann und damit einen deutlichen Beitrag leistet, die Time-to-Market von Software-Produkten zu reduzieren. Gleichzeitig schafft man durch die einhergehende Eliminierung ungeplanter Arbeiten Freiräume für Mitarbeiter und kann damit das Busi-

ness-Alignment deutlich ausbauen und die bestehenden neuen Anforderungen ausreichend flexibel erfüllen. Quasi nebenbei werden die natürlich existenten „Kopfmonopole“ („wer kann das mal konfigurieren, deployen, ...“) über bessere Verfahren abgeschafft und auch die Attraktivität zur Gewinnung neuer Mitarbeiter erhöht.



## GEMEINSAMES DENKEN UND ARBEITEN

Wenn die Anforderungen an Dev und Ops mit DevOps konsolidiert werden, dann beseitigt man auch Schmerzen, die sowieso auf der To Do-Liste stehen. So erhält die Organisation tatsächlich eine durchgängige Definition der Umgebung von Test auf Produktion, wenn es richtig gut gemacht wird auch mit aktuellen Daten, die der jeweiligen Umgebung entsprechend auch dem Datenschutz genügen. Weil Infrastruktur und Laufzeit als Code abgebildet werden und dadurch wiederholbar sind, kann sich dank der Automatisierung stark auf die Kernaufgaben fokussiert werden. Denn auch die Anwendungsumgebungen skalieren automatisch mit der Last und die sonst notwendigen manuel-

len Arbeiten entfallen auch hier. Besonders das Vor- und Zurückspielen von Änderungen an Anwendung und Infrastruktur wird beseitigt und führt zu besserer zielgerichteter Bereitstellung von Software an den Anwender und eliminiert das klassische Problemmanagement und Bugfixing. Wer Anwendungen auf Basis von Microservices anstrebt, hat kaum eine Alternative im Vorgehen, wenn diese Services autark entwickelt werden, damit dieses Modell überhaupt betreibbar wird.

Im Ergebnis sinken zunächst die Aufwände im Betrieb der Infrastruktur durch Automatisierung. Aber entfallen deshalb die Aufgaben, besser die

Arbeitsplätze? Ein deutliches „bedingt“ ist die Antwort. Die Aufgaben verändern sich, aber das Know-how bleibt weiterhin gefordert. Denn in den Entwicklungsteams muss das Wissen zu Netzwerk, zu Speicher, zu Betriebssystem – kurz zur Infrastruktur ja vorhanden sein. Die Anforderung an die Anwendung ist ja nun, diese mit zu definieren, zu testen und zu deployen. Das bedeutet dann, dass diese auch als Code definiert werden muss und hier ist das Entwicklungspotenzial für die Ops-Mitglieder eines DevOps Teams. Diese müssen ihr Wissen im Team einbringen – jetzt eben in einer anderen Lieferform, nämlich Code.

DevOps beseitigt bestehende Silos. Nicht die Arbeit in den Grenzen einer Abteilung, sondern direkte Kooperation ist gefragt. Besonders bei bereichsübergreifenden Effekten wie z. B. Requirements und deren Metriken sie zu messen. Das bedeutet auch gemeinsame Release-Planung, Architekturplanung oder Abstimmung zu den technischen Ressourcen, die eingesetzt werden sollen. Nukleus dieser Zusammenarbeit ist das autonome cross-funktionale Team, das minimal alle Development, QA- und Ops-Aspekte als Kompetenz und Entscheidungsfähigkeit enthält. Das Team kann selbstständig agieren und muss fachlich keine Rückfragen an Vorgesetzte stellen.

So wird es möglich, die benötigte Umgebung einfach auf Knopfdruck auszurollen und damit auch nachvollziehbare wiederholbare Infrastrukturen zu definieren, die Plattform-, Middleware- und Standardanwendungsservices zu integrieren und so zu geringeren Betriebsaufwänden führen, die gemanagt werden müssen.

Aber auch auf Dev-Seite braucht es dafür Bewegung. Gemeinsames Arbeiten im Team heißt auch, gemeinsames Verständnis über die Gesamtanforderung und damit den Input, die Anforderungen und den Mehrwert, den Ops in ein DevOps Team liefert.

Das Selbstverständnis eines DevOps Teams ist das Verständnis um Gemeinsamkeit im „Entwickeln“. Klassische Prozesse, Trennung in Aus- und Weiterbildung, organisatorische Strukturen, Standortsituationen etc. haben nicht selten dazu geführt, dass Gräben zwischen Dev und Ops entstanden sind und daher die Potenziale aus DevOps nicht gehoben werden können. Diese emotionalen Herausforderungen gilt es viel mehr zu managen, als die technischen. Letzteres kristallisiert sich zwar oft als Problem heraus, ist aber nie die Ursache. DevOps ist eben eine Art zu denken und zu handeln, die zwar durch Tools und Plattformen unterstützt wird, die aber durch die Menschen und deren Teamarbeit determiniert ist.



## EMPFEHLUNG

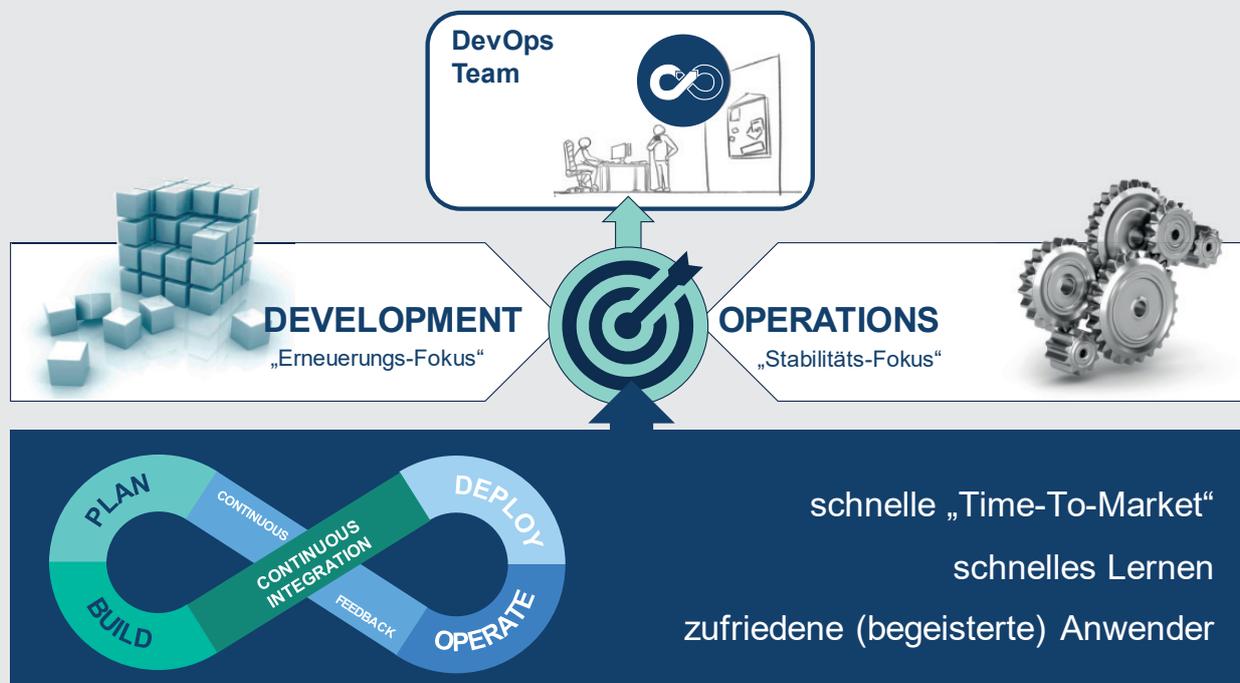
Weil DevOps ein Denken und Handeln eines Teams bezogen auf die Lieferung einer funktionsfähigen Software oder eines Services ist, sollten sich die Teams nicht auf den Betrieb der DevOps-Landschaft konzentrieren. Da die Plattformen – aus welchen Softwarekomponenten und deren Konfiguration auch immer – keinen Wettbewerbsvorteil darstellen, bietet sich der Betrieb der Plattform als Managed Service an. Ein guter Anbieter stellt die aktuellsten Versionen und Tools

für DevOps auch immer bereit und kümmert sich um das Thema Aktualität der DevOps-Landschaft.

Eine offene Architektur der Plattform erlaubt das Andocken an die im Unternehmen entweder schon genutzten Services oder Tools oder den Auf- und Ausbau solcher Komponenten in einem internen Projekt, das in diesem Modell dann aber nicht den Beginn eines DevOps-Zeitalters nach hinten verschiebt. Zudem lässt sich mit ei-

nem offenen Ansatz auch eine freiere Wahl bei der zu nutzenden Hardware realisieren, die besonders zum Start im Kontext Kosten punkten kann – quasi DevOps out of the Box.

Die Nutzung eines Managed Services für die eigentliche DevOps-Tool-Landschaft hilft, den Fokus auf die organisatorischen, kulturellen und teamorientierten Abläufe und Entwicklungen zu legen.



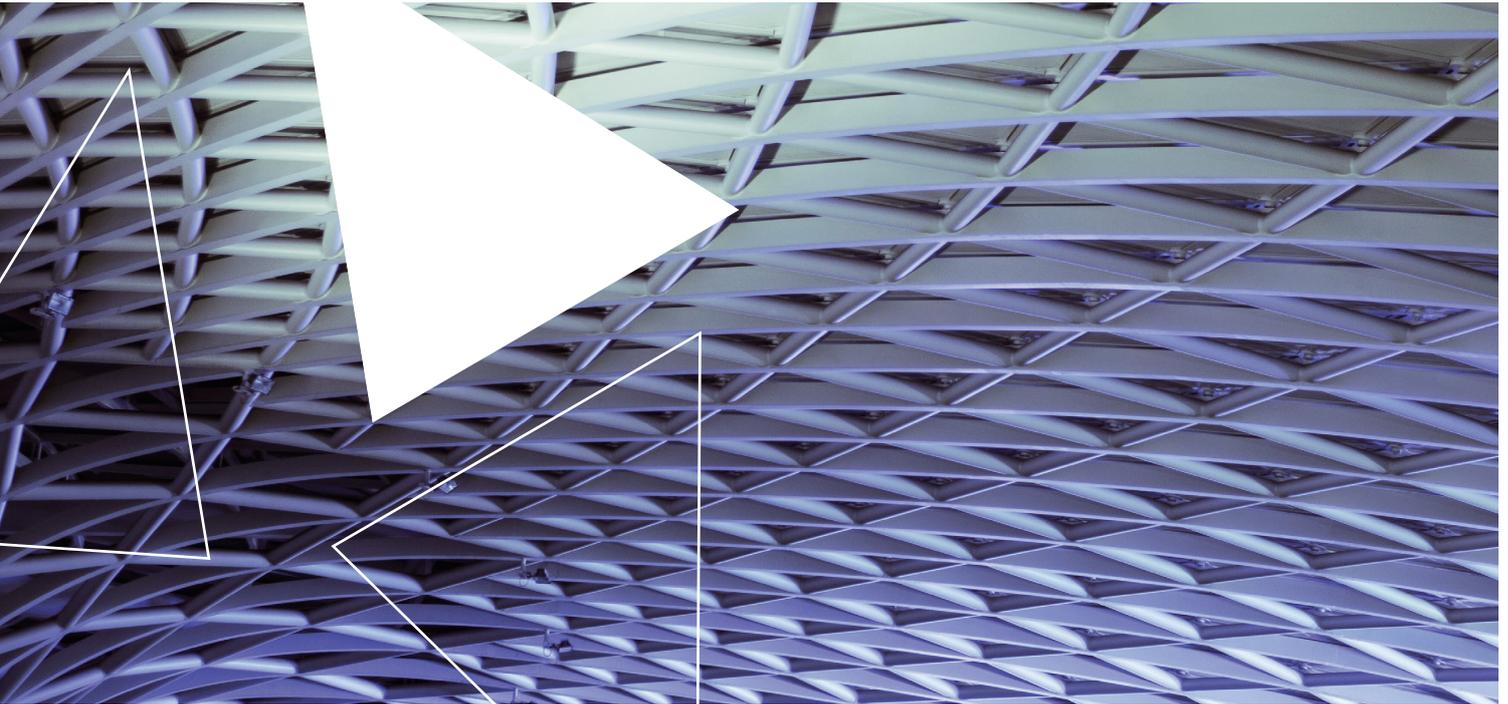
Da der Managed Service nicht davon abhängt, wie viele Builds und Deployments pro Tag durchgeführt werden, bleiben die Kosten für den Betrieb stabil. Anbieter, die die CI/CD Pipeline dynamisch skalieren („Built-Balancing“), helfen hier Hardware-Kosten zu optimieren und reduzieren Infrastrukturkosten durch die Möglichkeit eines Scale-Out von Spitzenlasten z. B. beim Testing in eine Cloud.

Ideal ist das Angebot als „Out-Of-The-Box“ Umgebung quasi auf Knopfdruck

zu erhalten. Das realisieren DevOps Managed Services Anbieter über das Deployment der Plattform per Code. So sind bei kurzfristiger Verfügbarkeit der Hardware von der Bestellung des Services bis Betriebsbereitschaft Lieferzeiten von unter einer Woche zu realisieren. Das ermöglicht bei kurzfristig angesetzten Projekten eine signifikante Reduzierung von Projekt Ramp-Up Zeiten und das mit genauer Skalierung auf den Bedarf des Kunden. Die Nutzung eines Managed Services für die eigentliche DevOps-Tool-

Landschaft hilft, den Fokus auf die organisatorischen, kulturellen und teamorientierten Abläufe und Entwicklungen zu legen.

Eine integrierte Kostenverrechnung für die benutzten Services je Projekt, ein Self-Service Portal inkl. APIs und harmonisierte Schnittstellen sollten selbstverständlich enthalten sein.



## ZUSAMMENFASSUNG

Nur wer sich im Klaren darüber ist, dass DevOps eine andere Art von Teamzusammenstellung und Vorgehensmodellen ist, wird tatsächlich die Ebene schneller Lieferungen zu attraktiven Kosten erreichen. DevOps kann man nicht anweisen, man muss die Organisation am besten über geeignete neue Projekte dahin entwickeln.

Wer DevOps in seiner DNA verankert hat, kann die Chancen in Bezug auf Kosten und Verfügbarkeit von Public und Private Cloud sowie OnPrem-Landschaft als koexistente hybride Infrastruktur tatsächlich grenzenlos nutzen.

DevOps Plattformen sind kein Asset, das Unterscheidungsmerkmale für die eigentlichen Anwendungen liefert – diese sind die Assets. Daher ist der Bezug der Plattform Managed Service eine praktikable Lösung, die Organisation selbst auf das Lernen und Umsetzen von DevOps zu fokussieren.

### Haben wir Ihr Interesse geweckt?

Dann sprechen Sie mich gerne an:

**Christoph Steinhauer**  
Geschäftsfeldleiter Agile  
Methoden & Digitalisierung  
[c.steinhauer@profi-ag.de](mailto:c.steinhauer@profi-ag.de)



# DIE PROFI ENGINEERING SYSTEMS AG

Wir sind ein mittelständisches, inhabergeführtes und finanzkräftiges IT-Lösungshaus mit Hauptsitz in Darmstadt. Innovationskraft und Kundenorientierung sind wesentliche Säulen unserer Unternehmensstrategie. Seit über 35 Jahren unterstützen wir unsere Kunden mit individuellen hochwertigen Lösungen zur Optimierung von IT-Prozessen und Systemlandschaften.

Wir begleiten Ihr Unternehmen bei der digitalen Transformation – von der Strategie über die Umsetzung bis zum Betrieb. Unsere erfahrenen Berater und Architekten beschäftigen sich seit vielen Jahren intensiv mit der Digitalisierung aller Geschäftsabläufe und Unternehmensbereiche. Gestalten Sie mit den PROFI-Fokusthemen schon heute Ihre digitale Zukunft. Profitieren Sie von unserem Know-how vor allem im Kontext von Managed Service Solutions, VDI & Digital Workplace, SAP HANA, Business Continuity, Agile Software-Entwicklung & DevOps, Netzwerk & Security, SDDC & Agile Plattformen, Speicherlösungen und Server-Lösungen. Wir übernehmen für Sie Projektmanagement und Implementierung, einschließlich dem Betrieb Ihrer Systeme und Plattformen. Unser Anspruch ist höchste Kompetenz, Zuverlässigkeit und Qualität, mit messbarem Erfolg und direktem Beitrag zur Wertschöpfung und Wettbewerbsfähigkeit unserer Kunden.

Wir beschäftigen rund 320 Mitarbeiterinnen und Mitarbeiter bundesweit an 12 Standorten. Seit vielen Jahren gehören wir zu Deutschlands erfolgreichsten Systemhäusern und pflegen langjährige Partnerschaften.

## Unsere IT-Lösungen für Ihren Erfolg

- Business Continuity
- Cloud & Hybrid
- DevOps & Agile Software-Entwicklung
- Managed Services & Managed Service Solutions
- SAP HANA
- SDDC & Agile Plattformen
- Security & Netzwerk
- Server-Lösungen
- Storage
- VDI & Digital Workplace



**PROFI Engineering Systems AG**  
Otto-Röhm-Straße 18  
64293 Darmstadt  
Telefon: +49 6151 8290-0  
Telefax: +49 6151 8290-7610  
E-Mail: [profi@profi-ag.de](mailto:profi@profi-ag.de)  
[www.profi-ag.de](http://www.profi-ag.de)

# UNSERE PARTNER

Gemeinsam mit unseren starken Partnern setzen wir Ihre optimalen Lösungen um.



12/2019

## Bildnachweise:

pexels.com:

Titel © Henry & Co