

DevOps: Survival-Training für den Public Sector

Der Zwang zur Optimierung der Delivery Pipeline als Auswirkung der digitalen Transformation

| von ERIK BENEDETTO und DR. ANDREAS ZAMPERONI¹

Software wird heute zunehmend agil entwickelt. Was noch vor wenigen Jahren als neue Vorgehensweise galt, wird zum De-facto-Standard in allen Branchen und für alle Unternehmensgrößen. Der Wunsch nach Geschwindigkeit, nach schnellen Ergebnissen in kurzen Sprints hat es ermöglicht, dass innerhalb eines kurzen Zeitraums agile Methoden die Softwareentwicklung substantziell verändert haben. Seit einiger Zeit geistert nun ein Begriff durch die Fachliteratur, der das Potenzial hat, die gesamte Prozesskette zu verändern – und zwar inhaltlich, strukturell und technisch: DevOps.

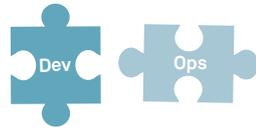
DevOps kann ein Bestandteil der Maßnahmen sein, um die digitale Transformation der traditionellen IT in der öffentlichen Verwaltung zu vollziehen und zu meistern. Aber was genau verbirgt sich dahinter? Wo liegt das Potenzial? Was sind die Bestandteile von DevOps, und wie lässt sich diese Methode in den Application Lifecycle integrieren?

In einer mehrteiligen Artikelreihe beleuchten wir alle Aspekte dieses Themas. Der erste Teil setzt sich mit folgenden Fragen auseinander:

- DevOps – Ursprung und Ziele?
- Wieso gewinnt DevOps im Kontext der digitalen Transformation eine neue Bedeutung?

¹ Dieser Artikel erschien ursprünglich in der msgGillardon NEWS und wurde für die .public von Dr. Andreas Zamperoni angepasst.

WAS IST DEVOPS?



DevOps ist ein Kunstwort, bestehend aus den Begriffen Development und Operations. Der seit 2008 verwendete Begriff setzt sich zusammen aus dem Wortbestandteil „Dev“, der die Softwareentwicklung (Development) repräsentiert, und „Ops“, der für den IT-Betrieb (Operations) steht. Die Kombination zu „DevOps“ symbolisiert intuitiv einen Schulterschluss zwischen Softwareentwicklung und IT-Betrieb. Und tatsächlich ist das der Grundgedanke von DevOps und der Auslöser der dazugehörigen Bewegung: ein Zusammenrücken der beiden, in der traditionellen Wahrnehmung grundverschiedenen Bereiche Softwareentwicklung und IT-Betrieb. Einerseits ist die Wortschöpfung DevOps sehr griffig. Andererseits birgt sie große Interpretationsspielräume, was zu Missverständnissen führen kann. Aktuell sieht die DevOps-Bewegung ihre Hauptaufgabe darin, die vielen Interpretationen zu kanalisieren und eine klare Definition von DevOps zu formulieren. Um zu wissen, warum dieser Schulterschluss notwendig ist, ist ein Verständnis des zugrunde liegenden Interessenkonflikts zwischen Dev (Entwicklung) und Ops (IT-Betrieb) nötig.

DEVELOPMENT: ZIELE DER SOFTWAREENTWICKLUNG

Ziel der Entwicklung ist die Komplettierung neuer Features durch schnelle und häufige Releases. Die Aufgabe von Softwareentwicklern besteht darin, die vom Auftraggeber gewünschten Funktionen und Features möglichst schnell umzusetzen. Durch die Entwicklung wird eine neue Funktion verfügbar, durch die sich ein potenzieller Mehrwert für die Endnutzer ergibt.

Je häufiger und schneller neue Features entwickelt und komplettiert werden, desto schneller kann dem Endnutzer also auch ein Mehrwert zur Verfügung gestellt werden. Gleichzeitig bedeutet ein schneller Entwicklungsprozess, zügig auf gesetzliche und Kundenanforderungen reagieren zu können. Daher wird dies in vielerlei Hinsicht positiv wahrgenommen – ebenso wie Entwickler, die diesen Anspruch erfüllen können.

Oft wird der Mehrwert einer Funktion allerdings schon bei der ersten Abnahme durch den Auftraggeber anerkannt und nicht erst dann, wenn diese dem Endnutzer letztendlich zur Verfügung steht. Der Grund hierfür ist einfach: Bis zum nächsten Release kann es schon mal ein paar weitere Tage dauern. Dadurch ist es allerdings für die Entwickler auch weitestgehend irrelevant, ob die neuen Funktionen/Features tatsächlich auf dem Produktionssystem verfügbar sind oder nicht.

OPERATIONS: ZIELE DES IT-BETRIEBS

Ziele des IT-Betriebs sind Stabilität und Sicherheit der Anwendungen und Infrastruktur durch wenige Releases bis hin zur Release-Vermeidung.

Die Aufgabe des IT-Betriebs besteht darin, die von der Entwicklung gelieferten Funktionen in Form von Software auf der Produktivumgebung für die Endnutzer verfügbar zu machen. Dazu zählen das Deployment im Rahmen neuer Software-Releases und gleichzeitig die Sicherstellung des laufenden Betriebs gemäß den definierten Qualitätsanforderungen und Rahmenbedingungen.

Der Betrieb trägt also die unmittelbare Verantwortung für die dauerhafte Verfügbarkeit der Anwendungen und deren Sicherheit. Der Erfolg wird daran gemessen, inwieweit die formalen Qualitätsanforderungen erreicht werden – meist in Form von definierten Service Levels und KPI, die in den Service Level Agreements dokumentiert sind. Da der Endnutzer in der Regel die volle Verfügbarkeit und Sicherheit der Anwendungen erwartet, ist es oberste Priorität des IT-Betriebs, diese im Rahmen der Service Levels sicherzustellen.

Aus diesen Gründen geht der IT-Betrieb entsprechend vorsichtig mit Veränderungen um. Abhängigkeiten zwischen verschiedenen Software-Deployments werden ausgiebig getestet und Deployments in Release-Paketen gebündelt. Insgesamt werden die Anforderungen an die Dokumentation und Tests sehr hoch angesetzt, bevor es überhaupt zu einem Deployment kommt. Servicebeeinträchtigungen sollen schnell beseitigt beziehungsweise von vornherein möglichst ausgeschlossen werden können.

Dies führt zu einer nachweislichen Entschleunigung der anfänglichen schnellen Softwareentwicklung. Die Gründe hierfür sind nachvollziehbar, denn ist die Verfügbarkeit einer Anwendung erst einmal beeinträchtigt, fällt das direkt auf den Betrieb zurück. Die Folge: eine stark negative Wahrnehmung durch die Auftraggeber, besonders wenn die Nutzer der Anwendung ein „Problem“ melden, noch bevor die verwendeten Monitoring-Systeme Alarm schlagen. Um die Wahrscheinlichkeit für unerwartete Ausfälle zu minimieren, setzt der Betrieb deshalb oft alles daran, den Zustand einer stabil laufenden Anwendung vor Änderungen zu schützen: durch wenige, gebündelte und ausgiebig getestete und dokumentierte Releases.

BLAME GAME: DER TRADITIONELLE KONFLIKT ZWISCHEN ENTWICKLUNG UND BETRIEB

Der Vergleich zeigt, dass beide Einheiten entgegengesetzte Anreize haben. Die Entwicklung ist an schnellen und häufigen Releases interessiert, der Betrieb hingegen würde Releases am liebsten vermeiden. Beide Seiten verfolgen damit jedoch das gleiche Ziel, nämlich ihren eigenen Wert für das Unternehmen oder die Behörde unter Beweis zu stellen.

Genau das führt aber regelmäßig zu Konflikten, denn in der Regel treffen Dev und Ops unter Zeitdruck aufeinander, zum Beispiel beim Deployment eines neuen Releases, oder wenn es ein Problem, wie beispielsweise einen Systemausfall, gibt. Dann beginnt oft das typische „Blame Game“, bei dem beide Lager sich gegenseitig die Schuld an der Situation geben.

Wer solche Situationen noch nicht erlebt hat, kann sich glücklich schätzen. Wer das Blame Game hingegen kennt, weiß: Schuldzuweisungen bringen nichts – die Zeit würde besser in die Lösung des Problems investiert.

Blame Game – zwei Beispiele

Die Entwicklung gibt ein neues Release zum Deployment an den Betrieb weiter, dem es jedoch nicht gelingt, die Software auf der Produktivumgebung lauffähig zu machen. Als der Betrieb die Entwickler kontaktiert und die auftretenden Fehler beschreibt, blocken diese ab: Die Software laufe auf der Entwicklungsumgebung fehlerfrei, der Fehler müsse beim Betrieb liegen. Beide Seiten schieben sich den schwarzen Peter zu. Nach Krisensitzungen und Unstimmigkeiten zwischen den Abteilungen ergibt eine Untersuchung, dass sich Entwicklungs- und Produktivumgebung in einem wichtigen Detail unterscheiden. Das war keiner der beiden Seiten vorher bewusst.

Im Produktivsystem taucht ein Performance-Problem auf. Unter großem Druck arbeiten die Entwickler mehrere Nächte durch und liefern schließlich einen Patch. Der Betrieb fürchtet jedoch, dass der Patch die Stabilität des Systems gefährden könnte, da er Änderungen an einer kritischen Komponente umfasst. Deshalb wird zunächst eine genaue Qualitätskontrolle auf einer Testumgebung verlangt, um die Lösung in realistischen Testszenarien zu überprüfen. Doch die benötigte Last lässt sich in der Testumgebung nicht adäquat darstellen. Einen Monat später ist der Patch noch immer nicht eingespielt. Die Entwickler sind enttäuscht, da „es ja offenbar doch nicht so eilig war“.

DER IT-BETRIEB ALS FLASCHENHALS

Seit die Softwareentwicklung verstärkt auf agile Methoden setzt, eskaliert die Situation immer häufiger, denn agile Methoden setzen auf kontinuierliche Interaktion zwischen Auftraggebern und Entwicklern sowie auf kurze Release-Zyklen. In der Regel setzt sich die damit einhergehende Philosophie aber nicht in den Betrieb fort. Die Vorteile agiler Methoden werden ausgebremst. Software-Releases werden zwar in kurzen Iterationen erstellt, der geschaffene Mehrwert wird aber erst viel später auf der Produktivumgebung sichtbar. Die immer kürzer werdenden Release-Zyklen offenbaren den Betrieb zunehmend als Flaschenhals auf dem Weg der Software zum Endnutzer. Außerdem erhöhen häufig stattfindende Releases das Potenzial für das direkte Aufeinandertreffen zwischen Softwareentwicklung und Betrieb.

Doch nun ist ein neuer Aspekt hinzugekommen, der den Handlungsbedarf besonders in der öffentlichen Verwaltung verschärft und schnelles Umdenken erfordert: die digitale Transformation.

DIE TRADITIONELLE UMSETZUNG VON GESETZEN UND VERWALTUNGSPROZESSEN UND IHRER IT-ARCHITEKTUR

Betrachtet man das bisherige „Geschäftsmodell“ und die IT-Architektur der traditionellen Fachabteilungen in den Behörden und ihrer Behörden-IT, stellt man fest, dass der Fokus bisher vor allem auf der Bereitstellung und dem Einsatz langlebiger Verfahren lag. Zunehmender Kostendruck erforderte jedoch eine Standardisierung der Verfahren. Dies hatte Auswirkungen auf die zugrunde liegende, zumeist monolithische IT-Architektur, um die Verfügbarkeit der Verfahren sicherstellen zu können. Host-Anwendungen und später klassische Datenbank- oder JEE-Anwendungen waren das Maß der Dinge und Basis der sogenannten Systems of Records.

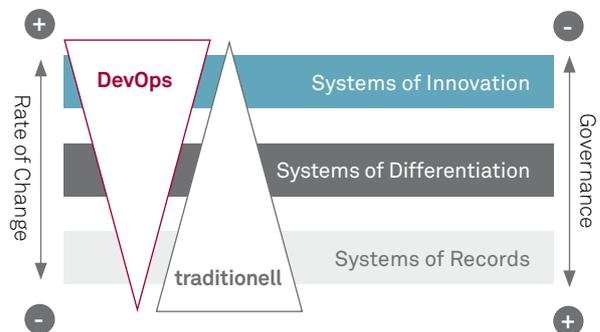


Abbildung 1: Fokus der Release-Strategien im Schichtenmodell der IT-Architektur, Quelle: Gartner

Release-Strategie	Fokus	Auswirkung
<ul style="list-style-type: none"> • Kurze Release-Zyklen (2–4 Wochen) • Flexible Konfiguration von neuen Produkten und Services 	<ul style="list-style-type: none"> • Garantiert Agilität und kurze Produkteinführungszeit 	<ul style="list-style-type: none"> • Unterstützt neue Dienstleistungen und Verfahren • Fördert Experimente und Innovation
<ul style="list-style-type: none"> • Mittlere Release-Zyklen (4–8 Wochen) • Verbindet Systems of Innovation mit Systems of Records 	<ul style="list-style-type: none"> • Interne Plattformentwicklung 	<ul style="list-style-type: none"> • Wiederverwendbare Querschnittsfunktionen • Standardisierte, hochqualitative Information
<ul style="list-style-type: none"> • Traditionelle Release-Zyklen (>2 Monate) • Stabile Funktion für das Kerngeschäft 	<ul style="list-style-type: none"> • Garantiert Stabilität und Kosteneffizienz 	<ul style="list-style-type: none"> • Unterstützt Verfahrenskontinuität, Konsolidierung und Compliance

Abbildung 2: Release-Strategien im Vergleich

Nur wenige Release-Termine pro Jahr waren erforderlich, um die IT-Architektur an die sich ändernden Rahmenparameter anzupassen. Noch heute sind zwei bis drei Releases pro Jahr der Standard in der öffentlichen Verwaltung.

DIE DIGITALE TRANSFORMATION IN DER ÖFFENTLICHEN VERWALTUNG VERSCHÄRFT DEN KONFLIKT

Durch die digitale Transformation steigt in der öffentlichen Verwaltung der Druck, den Konflikt zwischen Entwicklung und IT-Betrieb aufzulösen. Die Digitalisierung der Gesellschaft und der Wirtschaft sowie die Verfügbarkeit neuer Technologien haben die Anforderungen an die öffentliche Verwaltung hinsichtlich der Anpassungsgeschwindigkeit massiv erhöht.



Abbildung 3: Digitale Transformation – Status quo

Technologieunternehmen VS. öffentliche Verwaltung	
G+	5.500 pro Tag
f	3 pro Woche
🐦	1 pro Tag
2–3 /Jahr	

Abbildung 4: Deployment-Frequenzen der öffentliche Verwaltung im Vergleich

- Das geänderte Nutzerverhalten der Kunden als „Digital Natives“ erfordert neue Prozesse.
- Neue Technologien ermöglichen eine modulare, schnell skalierbare IT-Architektur aus der Box.
- Die Datenerhebung, -auswertung und -nutzung ermöglichen neue Geschäftsmodelle.
- Daten und das „Lernen aus den Daten“ werden zum integralen Bestandteil automatisierter Kommunikation und automatisierter Dienstleistungen.
- Zunehmender Druck durch gesellschaftliche und politische Veränderungen (zum Beispiel Flüchtlingskrise, Terrorismusbekämpfung) erfordern eine schnelle Umsetzung, um negative Konsequenzen für die Gesellschaft zu vermeiden.

Neue Anbieter von Dienstleistungen in allen Bereichen unserer Gesellschaft (zum Beispiel Apple, Amazon, Google)

- kommen aus dem technischen Umfeld, in dem sie die neuen Technologien definieren, entwickeln und betreiben,
- verfügen über große Datenmengen zum Nutzerverhalten und kennen sich mit den Kundenbedürfnissen durch detaillierte Datenauswertungen aus,

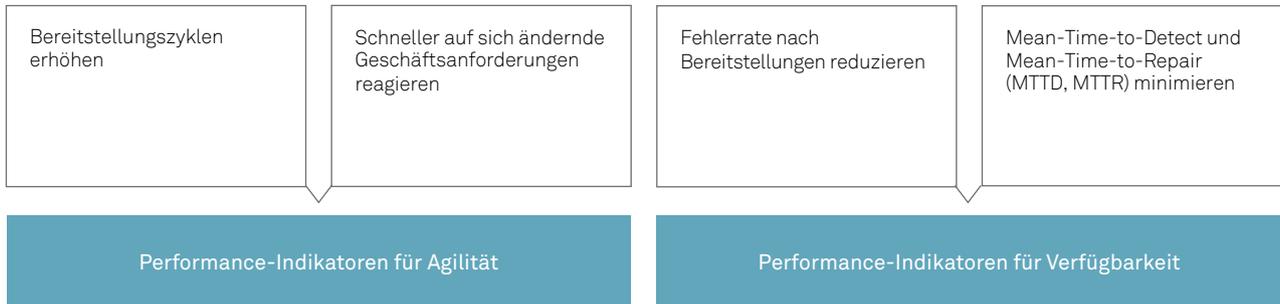


Abbildung 5: DevOps – Ziele und Metriken

- sind auf eine schnelle Time-to-Market getrimmt,
- haben schnelle, kurze Release-Zyklen, die durch eine modulare, integrierte IT-Architektur unterstützt werden,
- verfügen über hohe liquide Mittel aus ihrem Kerngeschäft

und erhöhen mit neuen und neuartigen Dienstleistungen den Druck auf die in der öffentlichen Verwaltung verfügbaren, traditionellen Dienstleistungsangebote für Bürger und Unternehmen, aber auch für Kunden innerhalb der öffentlichen Verwaltung selbst.²

DAS WETTRENNEN UM DIE KUNDEN UND GESELLSCHAFTLICHE RELEVANZ HAT BEGONNEN

Zusammengefasst war der Handlungsdruck in der öffentlichen Verwaltung nie größer, den Konflikt zwischen Softwareentwicklung und IT-Betrieb aufzulösen, denn er kostet Zeit, die die öffentliche Verwaltung für die digitale Transformation braucht,

- um die monolithische, transaktionsbasierte IT-Architektur in eine flexible, modulare umzubauen,
- um die Bereitstellungszeiten durch kurze Release-Zyklen drastisch zu erhöhen und damit den gesetzlichen und gesellschaftlichen Anforderungen zu genügen.

DevOps kann mit seinen Mechanismen helfen, die Delivery Pipeline, das heißt die Wertschöpfungskette, zu optimieren. Dabei werden zwei Bereiche unterschieden:

- Zusammenarbeit – Mechanismen, die den Konflikt zwischen Entwicklung und Betrieb auflösen, sowie
- Automatisierung und Prozesse – Erreichen einer hohen Release-Frequenz und -geschwindigkeit bei gleichzeitiger Verbesserung der Stabilität und Reproduzierbarkeit von Fehlern.

Der zweite Teil der Artikelreihe untersucht die Bestandteile und Mechanismen von DevOps und klärt dabei die Frage, wie sich DevOps in der öffentlichen Verwaltung implementieren lässt. ●

ANSPRECHPARTNER – DR. ANDREAS ZAMPERONI

Leiter Center of Competence
Projektmanagement
Public Sector Solutions Consulting



² Siehe Artikel „Fit für die nächste industrielle Revolution“ in diesem Heft.