

ITSM und Cloud(-native): Bewährtes weiterentwickeln

Wissensfrühstück „IT-Service-Check“

07.10.2026

Heiko Müller

Cloud(-native) verändert die **Bereitstellung** und **Steuerung** von **IT-Services**: **Prozesse** werden stärker **automatisiert**, Services standardisiert und Servicemodelle schlanker. Wir beleuchten, welche **ITSM-Praktiken** weiterhin tragen, wo **Anpassungen** notwendig sind und wie **Cloud-Steuerung, Automatisierung und Servicequalität** wirksam zusammenspielen.

„Wie lassen sich unsere Prozesse beschleunigen, ohne die Kontrolle zu verlieren?“

oder

„Warum dauert ein Change bei uns zwei Wochen, bei Google zwei Minuten?“



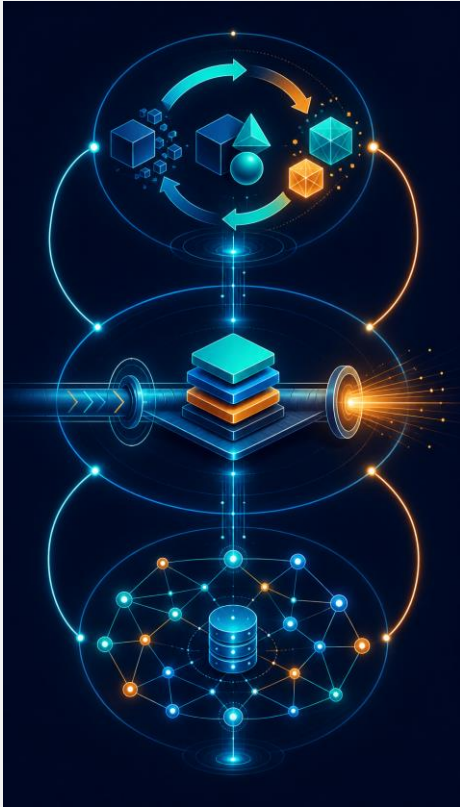
Heiko Müller – Senior Manager
DevOps und Cloud Transformation

A wide-angle photograph of a long cable-stayed bridge stretching across a calm body of water. The bridge features a series of tall, dark pylons supporting the deck with numerous stay cables. The sky is filled with soft, colorful clouds in shades of orange, pink, and blue, suggesting a sunset or sunrise. The water reflects the light from the sky and the bridge structure.

Agenda

1. Ausgangssituation und Herausforderungen

2. Beschleunigungsansätze



ITIL-Prozesse – was war nochmal wichtig?

- **Change Management**
 - Erkennen von Seiteneffekten
 - Nachvollziehbarkeit, Steuerung und Risikominimierung bei Änderungen an der IT
 - Vermeidung von (Human) Errors und Serviceunterbrechungen
- **Release Management**
 - Erkennen von Abhängigkeiten
 - Sorgfältige Planung und Bündelung von komplexen Änderungen
 - Fehlerfreies Steuern und Ausrollen der einzelnen Changes
- **Configuration Management**
 - Wissen über IT-Assets und deren Beziehungen aktuell halten
 - Integration mit Change Management notwendig



Wo klassische Prozesse typischerweise bremsen

- Change Advisory Board (CAB) als Flaschenhals
 - Manuelle, textbasierte Change-Anträge (keine CIs)
 - CMDB veraltet, weil manuell gepflegt
 - Release-Pakete groß, selten, riskant ("Big Bang")
- *Klassische Prozesse erreichen die definierten Ziele häufig unvollständig und sind zusätzlich langsam*



DevOps-Prinzipien als Hebel

- Automatisierung statt manueller Schritte
 - Wiederholbar und schnell
 - Infrastructure as Code (IaC)
 - Flexibel und iterativ testbar
 - CI/CD-Pipelines auf Git-Basis
 - Definition und Deployment gesichert und nachvollziehbar
 - Kleine, häufige Änderungen statt großer, seltener Änderungen
 - Fehlervermeidung, einfachere Fehlersuche und weniger Abhängigkeiten
- *DevOps ersetzt ITIL nicht, sondern liefert die Automatisierungs-Engine dafür*

Change Management mit GitOps neu gedacht

„Wenn ihr Lesezugriff für Eure Administratoren auf Eure Server erlaubt, ist Euer Monitoring kaputt – wenn Ihr sogar Schreibzugriff auf Eure Server erlaubt, ist zusätzlich das Change Management kaputt!“

(unbekannter DevOps-Verfechter)

- **Trennung von „Business-“ und „Technik-Anteilen“**
 - Warum und Wer (will es bzw. bezahlt es) -> Change Request
 - Wie und Wann -> IaC- und Git-basiert (inkl. Link/Task zum CR)
- **Parametrisierbare IaC-Skripte im Git mit CI/CD-Pipeline**
 - Flexibel anwendbar
 - Automatisiert, iterativ testbar (ersetzt manuelle Genehmigung)
 - Wiederverwendbar (auch wenn Ersteller nicht anwesend)
 - Nachvollziehbarkeit immer vorhanden
 - Manuelle Eingriffe eliminiert (über Rechte bzw. Re-Deployment)
 - Großer Hebel für BCM / Servicequalität

„Frequency Reduces Difficulty“

oder

„One of my favorite soundbites is:
if it hurts, do it more often.“

(Martin Fowler / Jez Humble)

Release Management - Frequenzerhöhung mit CI/CD

- **Von seltenen Großreleases zu häufigen kleinen Releases**
 - Herkömmliche Releasezyklen sind fehleranfällig, schwerfällig und schlecht aus Sicht „Time-to-Market“
 - Automatisierung erzeugt schnelles Feedback, senkt die „Batch Size“ und somit Risiko und Downtimes
 - Verbindung von ITSM- und CI/CD-Tools in Release-Workflows
- **fehlertolerante Deployment-Techniken**
 - Blue-Green- und Canary-Deployments -> zusätzliche, automatisierte Reaktion in Fehlerfällen
 - Feature Flags -> Entkopplung von Release und Deployment (dezentrale Szenarien)

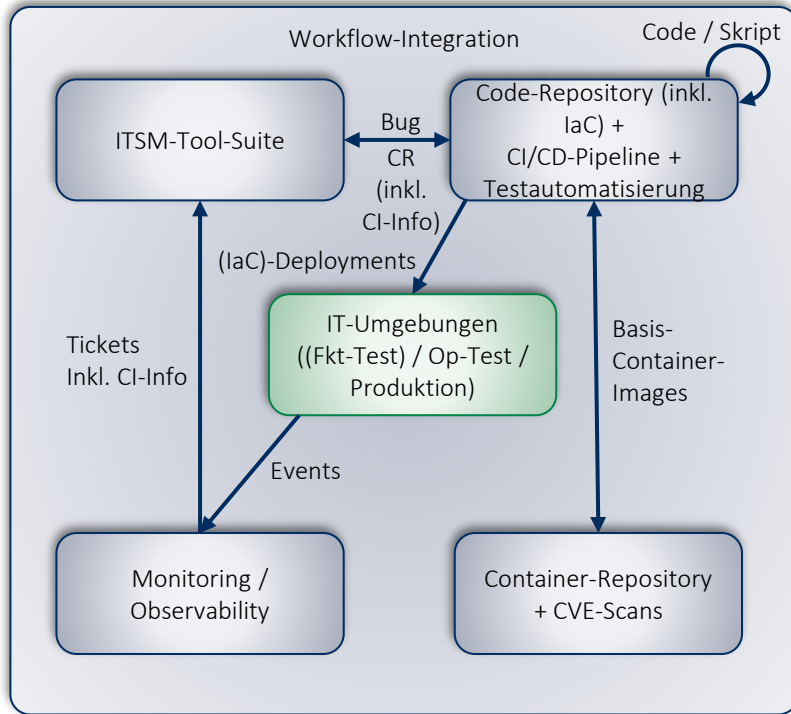
„Cattle, no Pets“

(Randy Bias)

Configuration Management inkludiert

- **Automatisierung der CI-Änderungen**
 - Über GitOps werden die CI-Änderungen „mitautomatisiert“ statt manuell gepflegt (Fehlervermeidung)
 - Änderungen sind Bestandteil des Codes (inkl. Tags z. B. zur Verrechnung)
 - Soll-/Ist-Abgleich wird überflüssig
 - In Cloud-Umgebungen sind die (technischen) Config-Daten immer vorhanden (abrechnungsbedingt)
 - **Notwendige CI-Abbildungstiefe verringert sich häufig über verbesserte Resilienz**
 - Auto-Scaling
 - Immutable Infrastructure
 - Verhinderung von Configuration Drift
 - Potenzieller Einsatz von Chaos Monkey

Welche Tools brauchen Sie, um diese Integration zu erzeugen?



Fazit

- **Automatisierung** reduziert manuelle Arbeit in allen drei Prozessen und erhöht somit Geschwindigkeit (letztendlich Time-to-Market)
- **ITIL-Ziele** (Kontrolle, Nachvollziehbarkeit) bleiben erhalten – nur der Weg dahin wird (weitgehend) automatisiert
- **Kulturwandel** nötig: Vertrauen in Pipelines statt in manuelle Freigaben





FRAGEN UND DISKUSSION



Schloßstraße 1 | 12163 Berlin

info@hisolutions.com | +49 30 533 289 0

www.hisolutions.com