



# Steuerung in Zeiten von DevOps

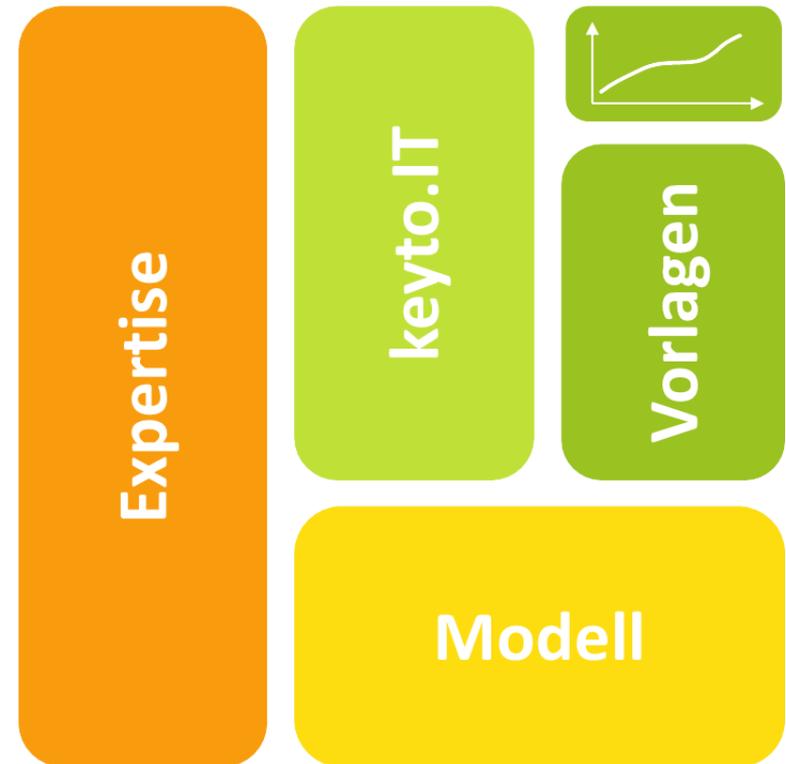
## Ziele und Kennzahlen

Dr. Andreas Knaus

# LINJAL GmbH



- LINJAL GmbH liefert Lösungen und Beratung zur Steuerung von Service Providern:
  - Portfolio
  - Steuerungsprozesse
  - Performanceoptimierung
- Gegründet: 02.01.2014  
Standort: München
- Geschäftsführer:  
Dr. Andreas Knaus





## Ganzheitliche Beratung, Analyse und Software zur Steuerung von IT-Dienstleistungen



Preise und Kosten transparent gestalten



Prozesse, vom Vertrieb bis zur Delivery, optimieren



Technologie wertschöpfend nutzen



Portfolio bedarfs- und marktgerecht gestalten

für interne und externe IT Service Provider



# Agenda

- Betriebsübergabe – Herausforderung zwischen Entwicklung und Betrieb
- DevOps – zwischen Management und Kollaboration
- Kennzahlen und Steuerung – Ziele definieren und Leistung messen



## **Betriebsübergabe**

**Herausforderung zwischen Entwicklung und Betrieb**



# Digitalisierung / Industrie 4.0

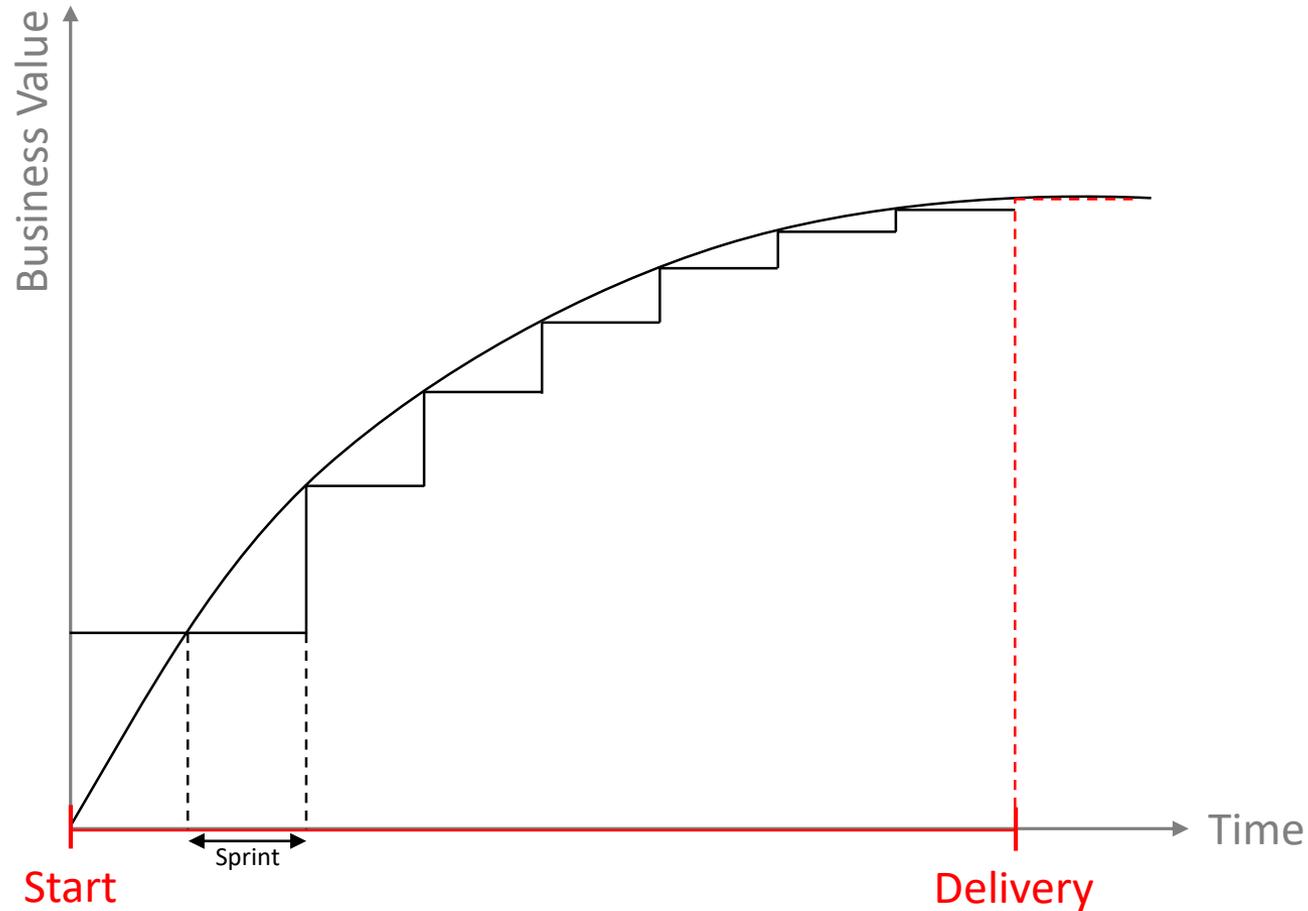
- Zunehmende Nutzung von Informationstechnologien
  - Potentiale
    - Neue Geschäftsfelder
    - Verbesserung der Wertschöpfung
    - Erhöhung der Vermarktungsreichweite
    - Senkung der Transaktionskosten
  - Nutzung
    - Unternehmensintern
    - Vermarktung
      - B2C
      - B2B
    - Zusammenarbeit zw. Unternehmen (über die gesamte Lieferkette)



# Herausforderung

- Time to Market
  - Hohe Geschwindigkeit bei der Entwicklung neuer, softwaregetriebener Services
  - Hohe Zyklusgeschwindigkeit bei der Umsetzung stellt besondere Anforderungen an die Qualitätssicherung
  - Variantenreichtum erhöht Aufwände bei der Qualitätssicherung
  - Sicherheit bekommt wegen der starken Datenorientiertheit mehr Gewicht
- Veränderung der Plattform
  - Consumer-orientiert
  - Anpassung der technischen Basis / Architektur
  - Einbindung von Partnern (Sourcing, Cloud, ...)

# Agilität, Qualität, Geschwindigkeit



# Transition



## Service Management

### Service Strategy

- Strategy
- Portfolio
- Financial
- Demand
- Business Relation

### Service Design

- Koordinierung
- Service Catalogue
- Service Level
- Risiko
- Capacity
- Availability
- Service Continuity
- Information Security
- Compliance
- Architecture
- Supplier

### Service Transition

- Change Mgmt
- Change Evaluierung
- Projektmgmt
- Anwendungs-entwicklung
- Release Deployment
- Service-Validierung Test
- SACM
- Knowledge

### Service Operations

- Event
- Incident
- Request Fullfillment
- Access
- Problem
- IT Operations Control
- Facility
- Application
- Technical

### Continual Service Improvement

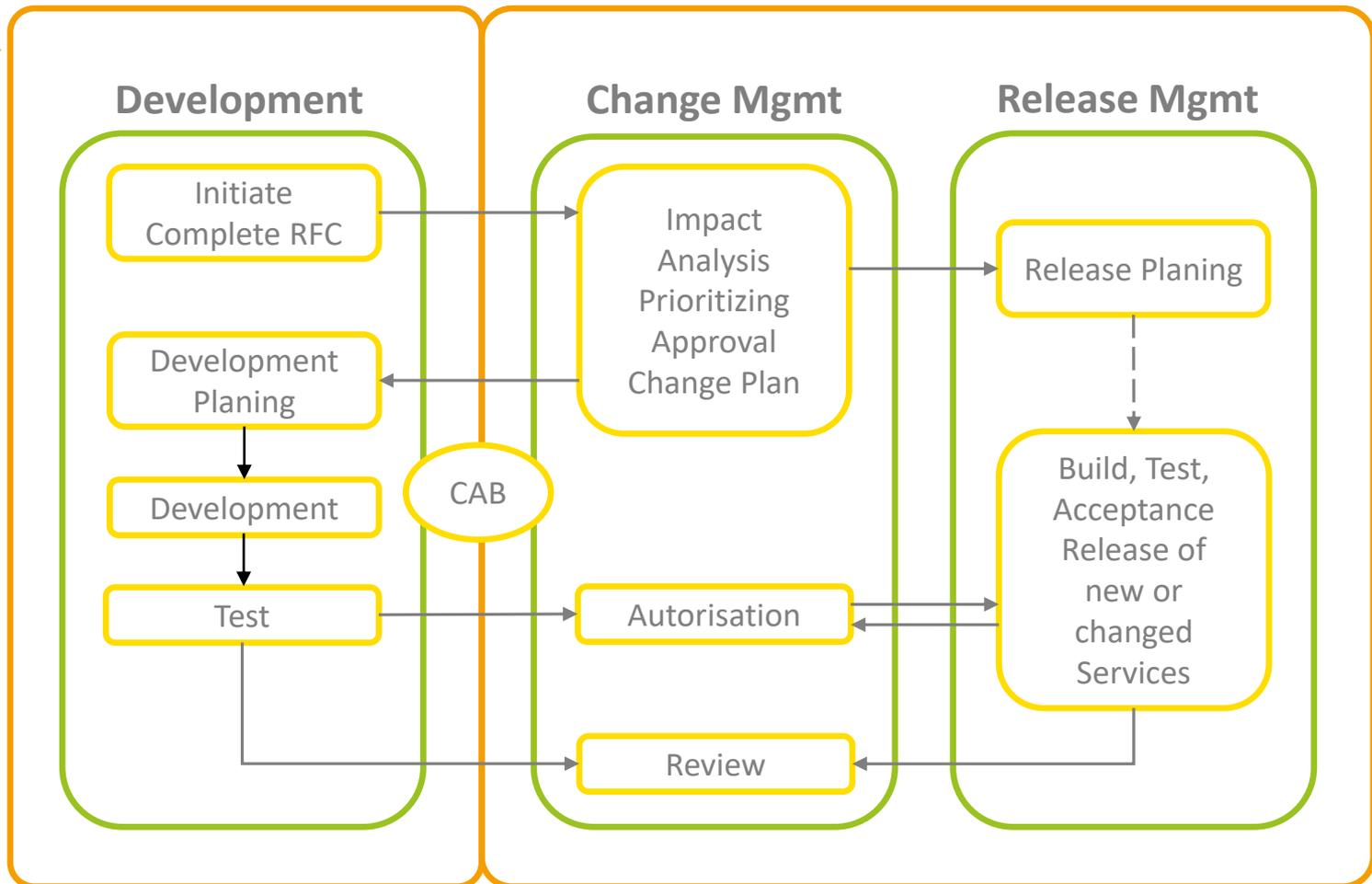
- Service Review
- Prozess Evaluierung
- Definition von CSI-Initiativen
- Überwachung von CSI-Initiativen



# Release & Control

Projekt / Technical Support

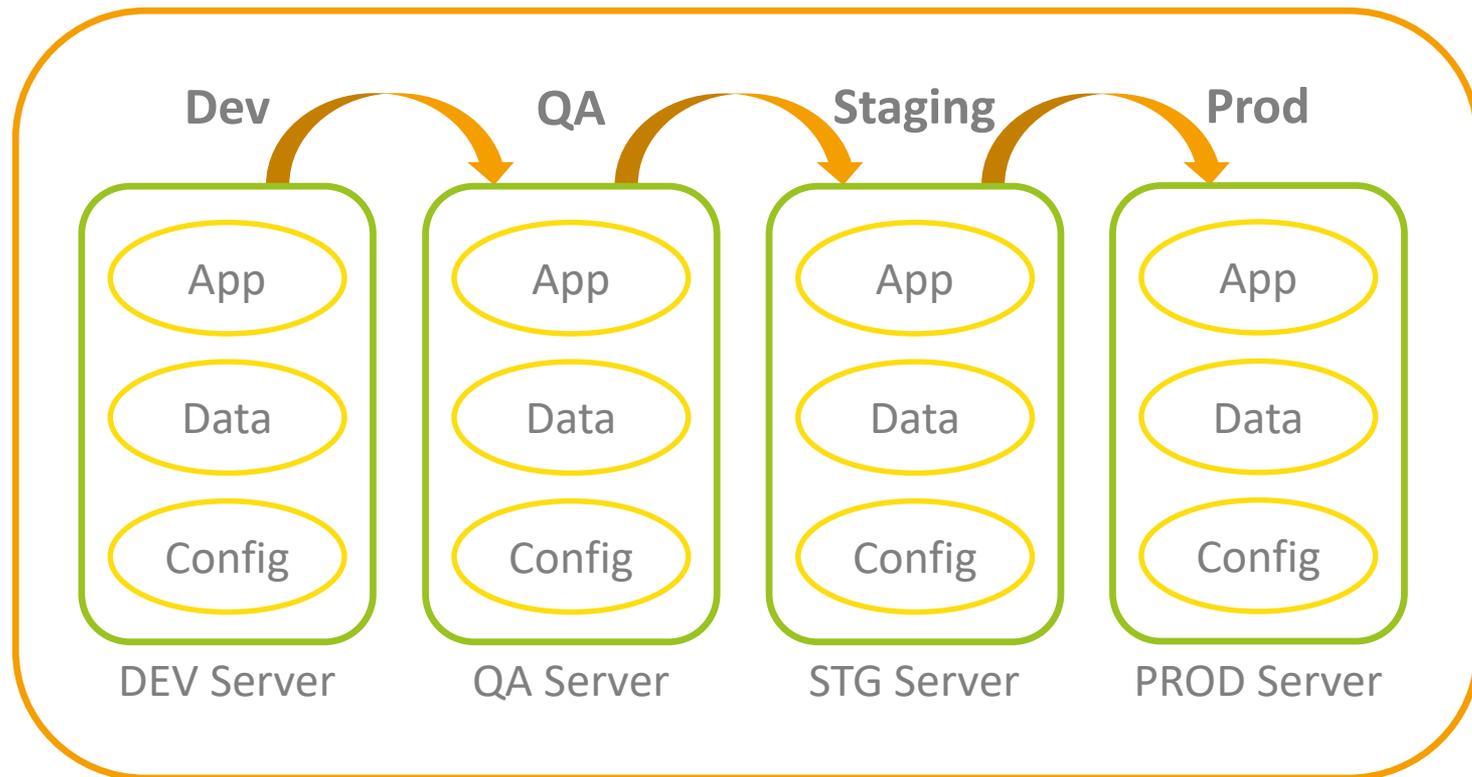
Service Management / IT Operations







# Service Management Release - Staging





## DevOps

**Zwischen Management und Kollaboration**



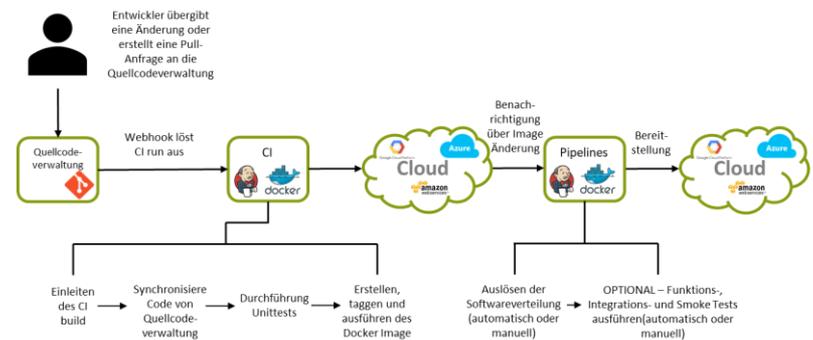
# Lösung aus Sicht der IT

- Dynamische Bereitstellung der Leistung
  - Agile Entwicklung
  - Nutzung von Cloudtechnologien
- Details
  - Produktmanagement
    - Organisation der Produktvarianten über den Lebenszyklus
  - Architektur
    - Isolierung von Funktionalität in Services
    - Trennung von Daten
    - Konfiguration als Configurations Items (Service Management)
  - Projektmanagement
    - Organisation der Anforderungen über den Lebenszyklus
  - Entwicklungsprozess und Qualitätssicherung
    - Parallele Entwicklung
    - Testintegration
    - Testautomation
  - Operations
    - Schnelle Bereitstellung von passenden Ressourcen und Umgebungen

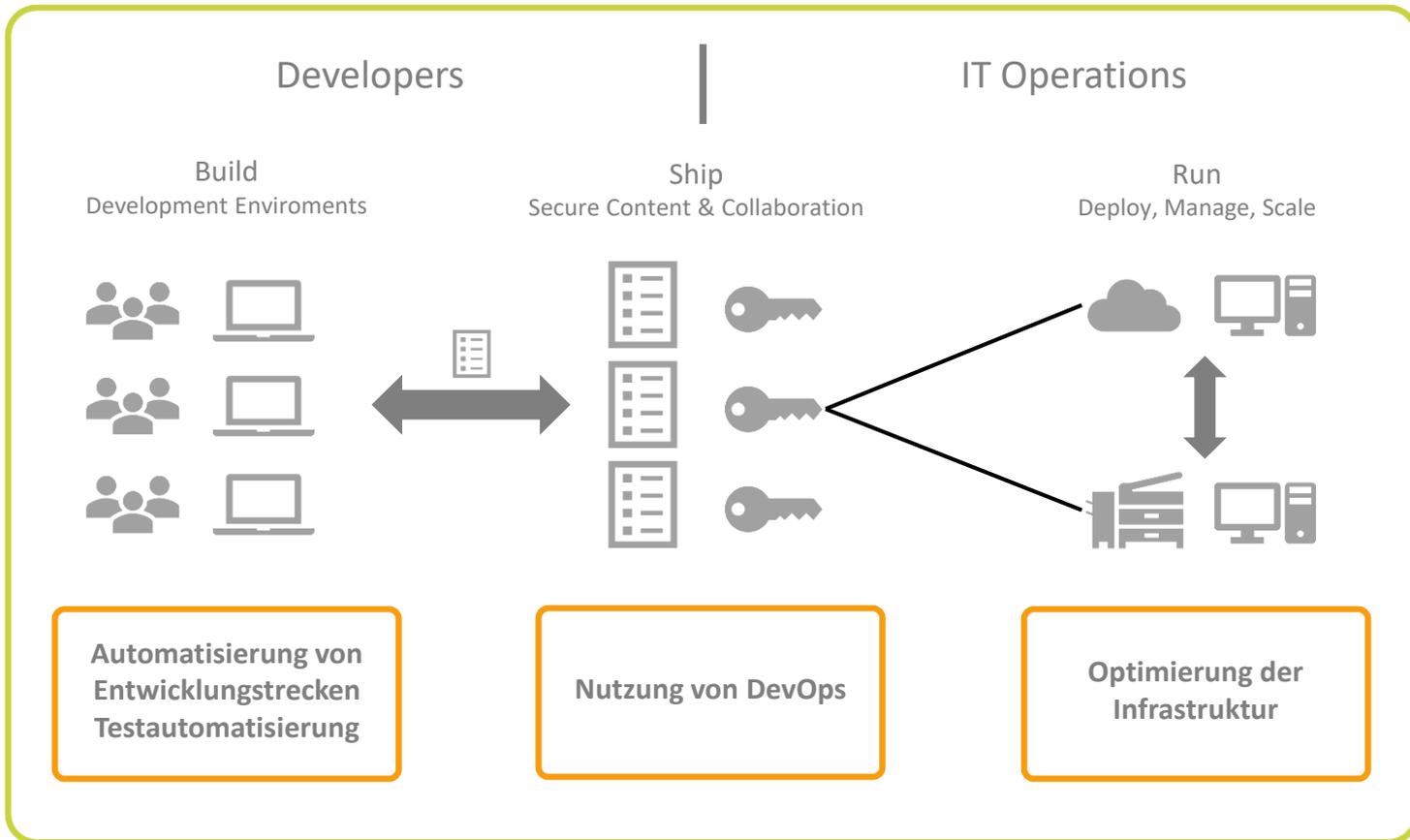
# Continuous Integration und Delivery



- Ziel: Verbesserung des Softwareauslieferungsprozesses
  - Themen
    - Continuous Integration
    - Testautomation
    - Automatisches Deployment
  - Vorteile
    - Schnelle Auslieferung
    - Zuverlässigkeit
    - Wiederholbarkeit
- ➔ Erweiterungen und Fehlerkorrekturen können schneller und risikoärmer ausgeliefert werden



# DevOps



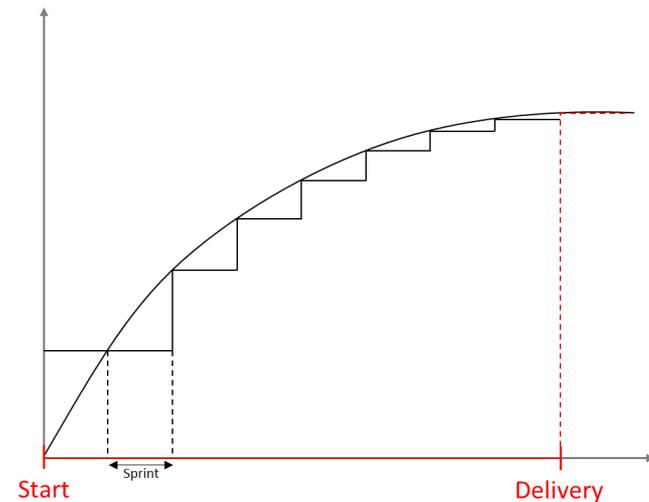


# DevOps

- Ziel: ganzheitliche Verbesserung

- Inhalte

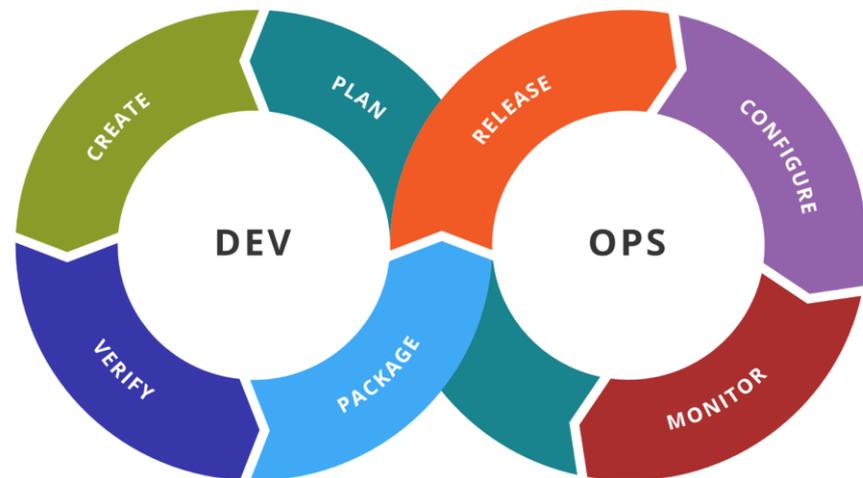
- integrative Prozesse
- passende Technologien
- hohe Automation
- kooperative Zusammenarbeit der Mitarbeiter





# DevOps – Grundsätze

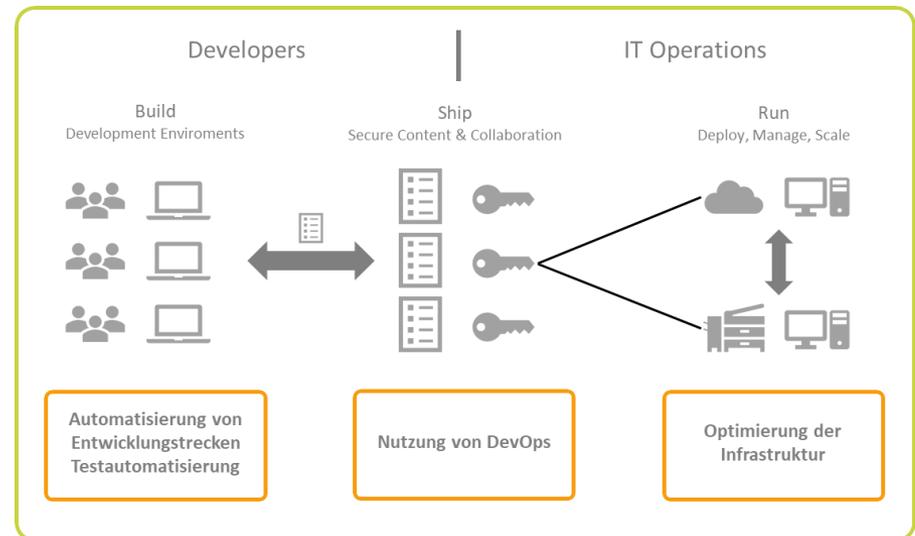
- Holistisch
- Kollaborativ
- Iterativ
- Inkrementell
- Kontinuierlich
- Automatisiert
- Self-Service





# DevOps – Prozesse

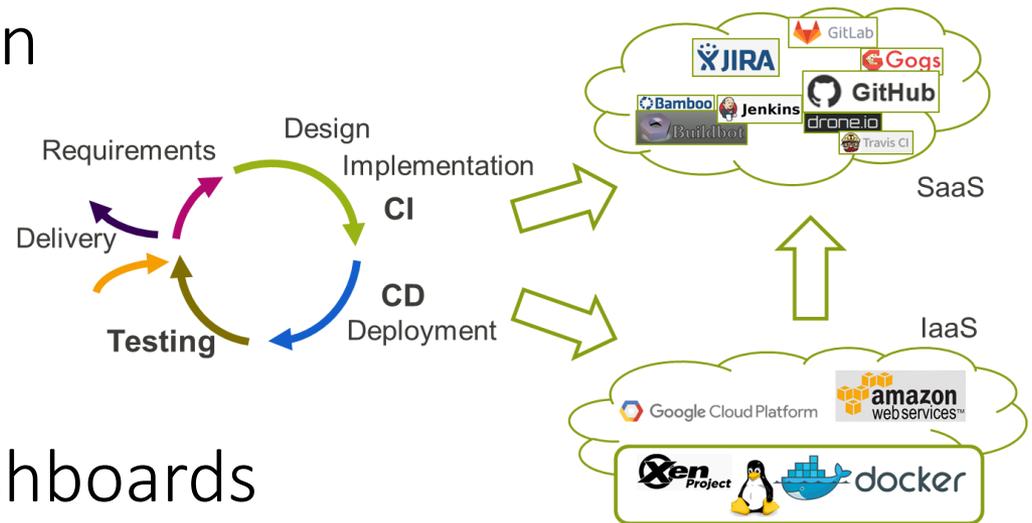
- Übergreifende Steuerung
- Anforderungsaufnahme
- Entwurf und Design
- Entwicklung und Testing
- Deployment
- Support
- Monitoring und Optimierung





# DevOps – Technologien

- Requirements Tracking
- Version Control & Configuration Management
- Build & Integration
- Orchestration
- Test
- Deployment
- Monitoring & Dashboards
- Support (Events, Issues & Tickets)





# DevOps – Mitarbeiter und Team

- Mitarbeiter
  - Soziales Miteinander
  - Kollaborative Zusammenarbeit
- Team-Kompetenzen
  - Design
  - Entwicklung
  - Qualitätsmanagement
  - Betrieb
  - Geschäftspartner / Fachliche Nutzer



# Kennzahlen und Steuerung

**Ziele definieren und Leistung messen**



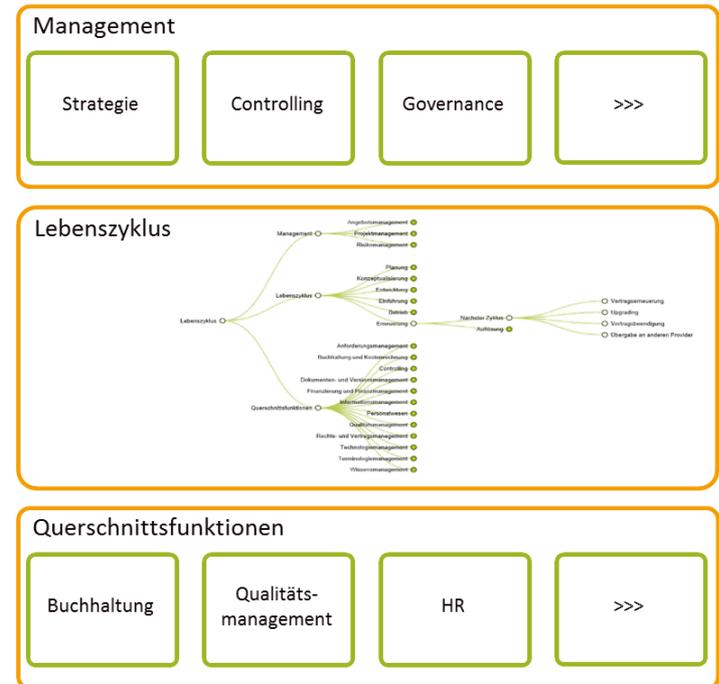
# Unternehmensziele





# Service Lebenszyklus

- Vollständiger Servicelebenszyklus
  - für Dienstleister
  - für Kunden
- Ca. 200 Aktivitäten
- Übergreifende Inhalte
  - Managementaufgaben
  - Querschnittsfunktionen
- Grundlage für Kostenmodell

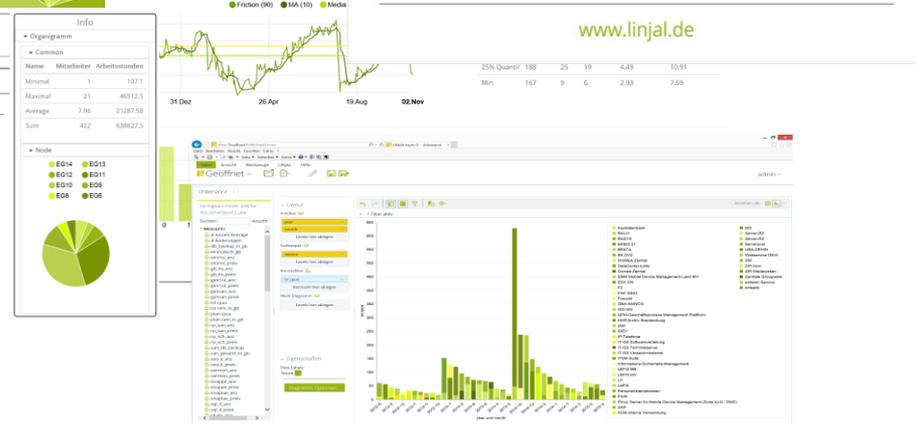
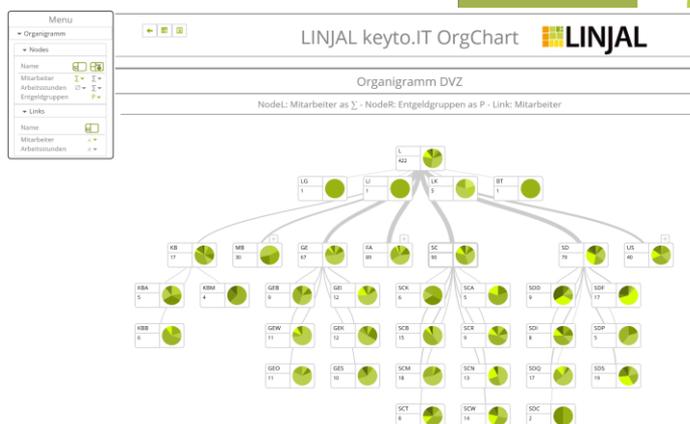
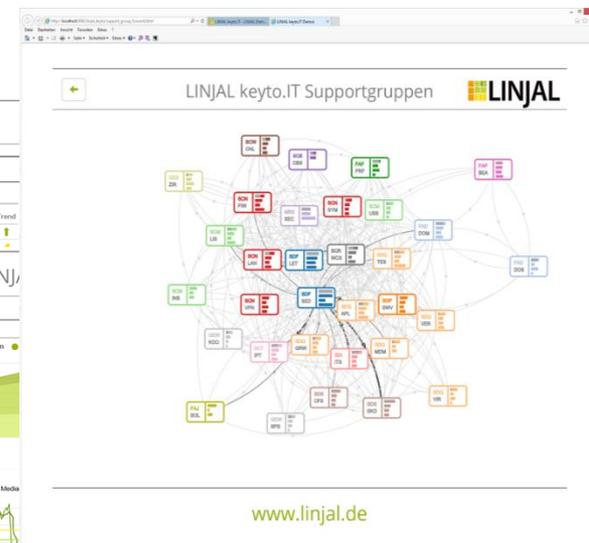
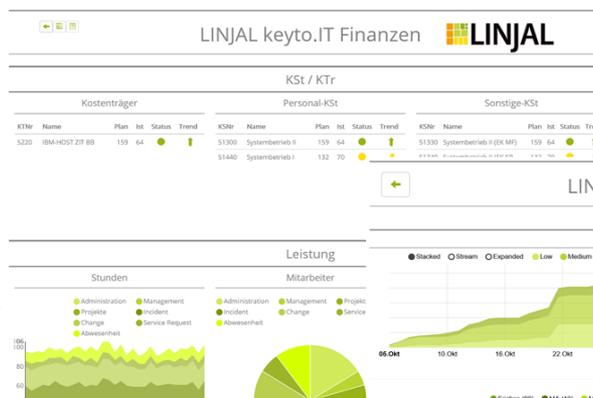


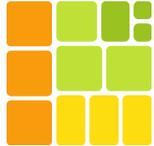


# Kennzahlen und Dashboards

## Steuerungsrelevante Informationen

- Kennzahlen
- Dashboards
- Reports
- Ad-Hoc-Analyser





# Kennzahlenkatalog

<http://www.linjal.de/kennzahlenkatalog/>

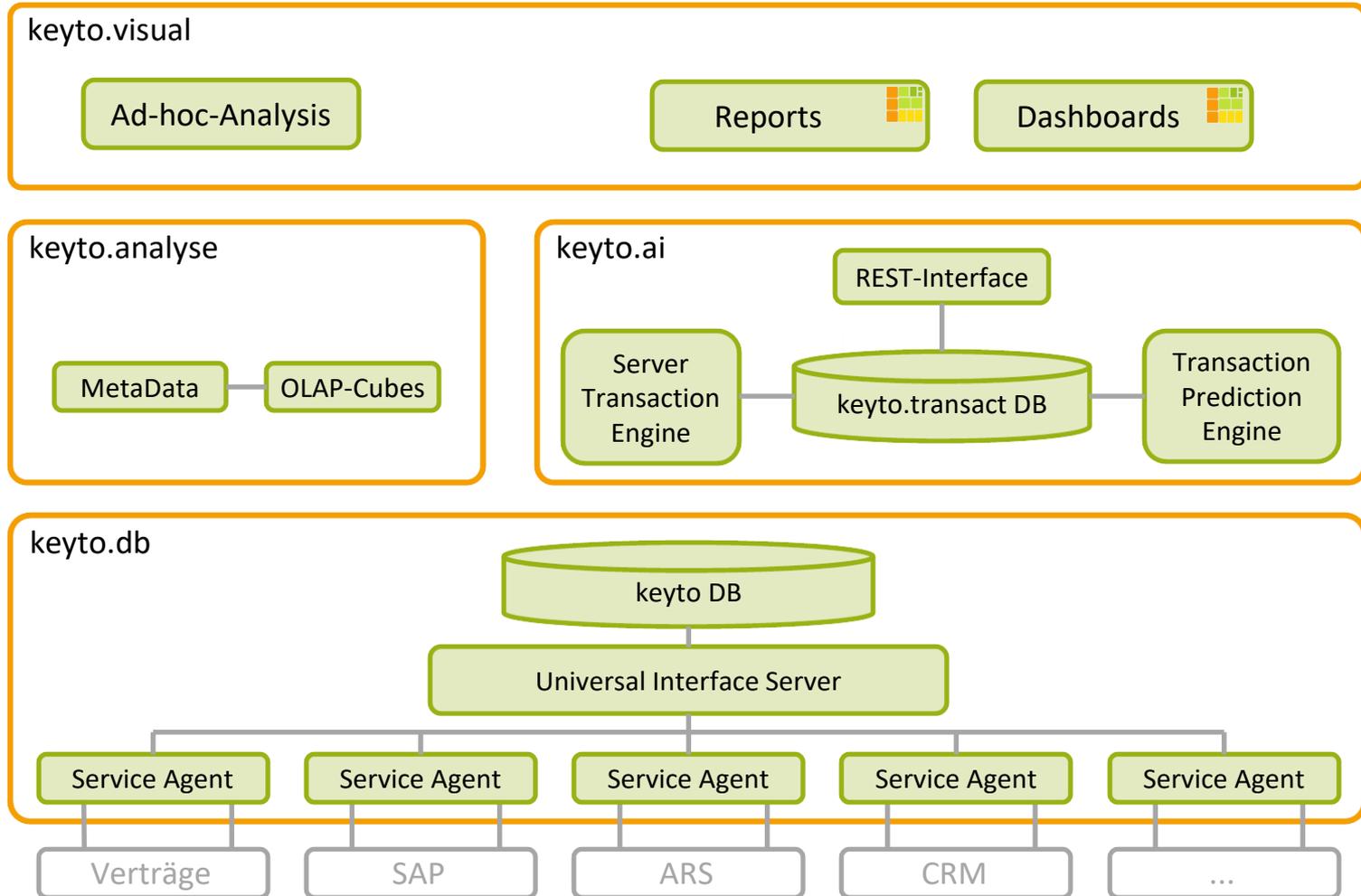
## Bereitstellungskostenanteil

Name	Bereitstellungskostenanteil share of provisioning costs	Bereich	Betrieb
Typ		Trendzahl	Verhältniszahl
<b>Beschreibung</b>	Die Kennzahl drückt den Anteil der Kosten am Umsatz oder Nutzen aus, der durch die Bereitstellung einer Leistung entsteht. Sonstige Kosten, wie etwa Investitionen oder laufende Kosten bleiben unberücksichtigt. Damit liefert die Kennzahl einen Indikator für die Effizienz des Bereitstellungsprozesses. Wird beispielsweise eine virtuelle Maschine bereitgestellt und nur kurz für einen Test benötigt, so ist damit zu rechnen, dass der Bereitstellungskostenanteil relativ hoch ist. Hier ist ein hohes Maß an Automation notwendig, um Effizienz zu erzielen. Wird die virtuelle Maschine dagegen langfristig in der Produktion genutzt ist der Bereitstellungskostenanteil erwartungsgemäß sehr gering.		
<b>Formel</b>	Externe Nutzung: $\text{Bereitstellungskostenanteil} = \frac{\text{Kosten für die Bereitstellung}}{\text{Umsatz durch Nutzung}} \cdot 100\%$ Interne Nutzung: $\text{Bereitstellungskostenanteil} = \frac{\text{Kosten für die Bereitstellung}}{\text{Mehrwert durch Nutzung}} \cdot 100\%$		
<b>Häufigkeit</b>	Quartalsweise Jährlich		
<b>Abgrenzung</b>	$\text{Kostenanteil} = \frac{\text{Gesamtkosten}}{\text{Umsatz}} \cdot 100\%$ $\text{RoI} = \frac{\text{Gewinn}}{\text{Umsatz}} \cdot 100\% = \frac{\text{Gewinn}}{\text{Gesamtkapital}} \cdot 100\%$		
<b>Varianten</b>	bezogen auf: Infrastruktur Services		
<b>Beispiel</b>	Eine virtuelle Maschine wird für eine Testumgebung benötigt. Im Rahmen des Bereitstellungsprozesses fallen Kosten in Höhe von 20 € an. Der Preis für die Nutzung der Maschine für zwei Wochen beträgt 400 €. Bereitstellungskostenanteil = 20 € / 400 € = 5%		
<b>Typische Werte</b>	Bei produktiven Umgebungen sollten die Bereitstellungskosten im niedrigen Promillebereich und darunter liegen. Auch in volatilen Umgebungen sollten einstellige Prozentbereiche nicht überschritten werden.		
<b>Anwendung</b>	Fachlich <ul style="list-style-type: none"> <li>• Prüfung der Effizienz der Bereitstellung</li> <li>• Indikator für die Notwendigkeit und auch Sinnhaftigkeit einer Automation</li> </ul> Organisatorisch <ul style="list-style-type: none"> <li>• Betriebsprozesse</li> <li>• Change Management</li> <li>• Service Level Management</li> </ul>		

## Anteiliger Aufwand nach Umsetzungsphasen

Name	Anteiliger Aufwand nach Umsetzungsphasen expenditure ratio by phase	Bereich	Softwareentwicklung
Typ		Verhältniszahl	
<b>Beschreibung</b>	Die Kennzahl teilt die Gesamtaufwände bei der Entwicklung einer Software in die typischen Projektphasen (Analyse, Design, Entwicklung und Qualitätssicherung) auf. Dabei ist es unerheblich, ob die Arbeiten sequentiell oder iterierend erfolgen. Mit Hilfe der so gewonnenen Informationen werden Abschätzungen gestützt und Benchmarks durchgeführt.		
<b>Formel</b>	$\text{Aufwand nach Umsetzungsphase} = \frac{\text{Aufwand der Phase}}{\text{Gesamtaufwand}} \cdot 100\%$		
<b>Häufigkeit</b>	Nach Abschluss der Entwicklung		
<b>Abgrenzung</b>			
<b>Varianten</b>	bezogen auf: Arbeitsstunden Personalkosten (intern und extern) Zeitverbrauch Investitionen		
<b>Beispiel</b>	Bei einem Gesamtumsetzungsvolumen von 1000 Stunden wurden 400 Stunden auf die eigentliche Entwicklung verwendet. Damit entspricht der anteilige Aufwand für diese Phase 400/1000 * 100% = 40%.		
<b>Typische Werte</b>	Die konkrete Aufteilung der Aufwände bei den Phasen hängt stark von der Reife und vom Standardisierungsgrad ab. Trotzdem ist in der Regel eine Aufteilung von <ul style="list-style-type: none"> <li>• Analyse: 10%</li> <li>• Design: 20%</li> <li>• Implementierung: 30%</li> <li>• Qualitätssicherung: 30%</li> </ul> zu erwarten.		
<b>Anwendung</b>	Fachlich <ul style="list-style-type: none"> <li>• Indikator bzw. Referenz für Richtigkeit der Projektabschätzungen</li> <li>• Benchmarking</li> </ul> Organisatorisch <ul style="list-style-type: none"> <li>• Projekt</li> <li>• Projektcontrolling</li> <li>• Projektmanagement</li> </ul> Softwareentwicklung <ul style="list-style-type: none"> <li>• Entwicklungsleitung</li> </ul>		

# keyto.IT Architektur





# Kontakt

**Dr. Andreas Knaus**

Landwehrstr. 61

80336 München

[aknaus@linjal.de](mailto:aknaus@linjal.de)

01523 1860455