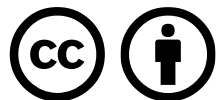


# DevOps

Flow (Forts.)

Andreas Heil



Dieses Werk ist lizenziert unter einer Creative Commons  
Namensnennung 4.0 International Lizenz.

# Lernziele

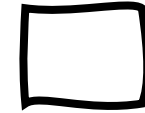
**Verstehen** warum viele Übergaben zu langen Deployment-Durchlaufzeiten führen.

**Verstehen** wie man Flaschenhälse identifiziert und Ansätze **kennen lernen**, diese zu beseitigen.

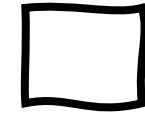
Das Konzept der Verschwendung **kennen lernen** und **verstehen** wie man Verschwendung vermeidet.

# Arbeitsschritte

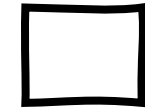
Wie viele Schritte halten Sie in einem Prozess für realistisch, um Code aus der Versionsverwaltung in die Produktivumgebung zu bringen?



1-10



10-100



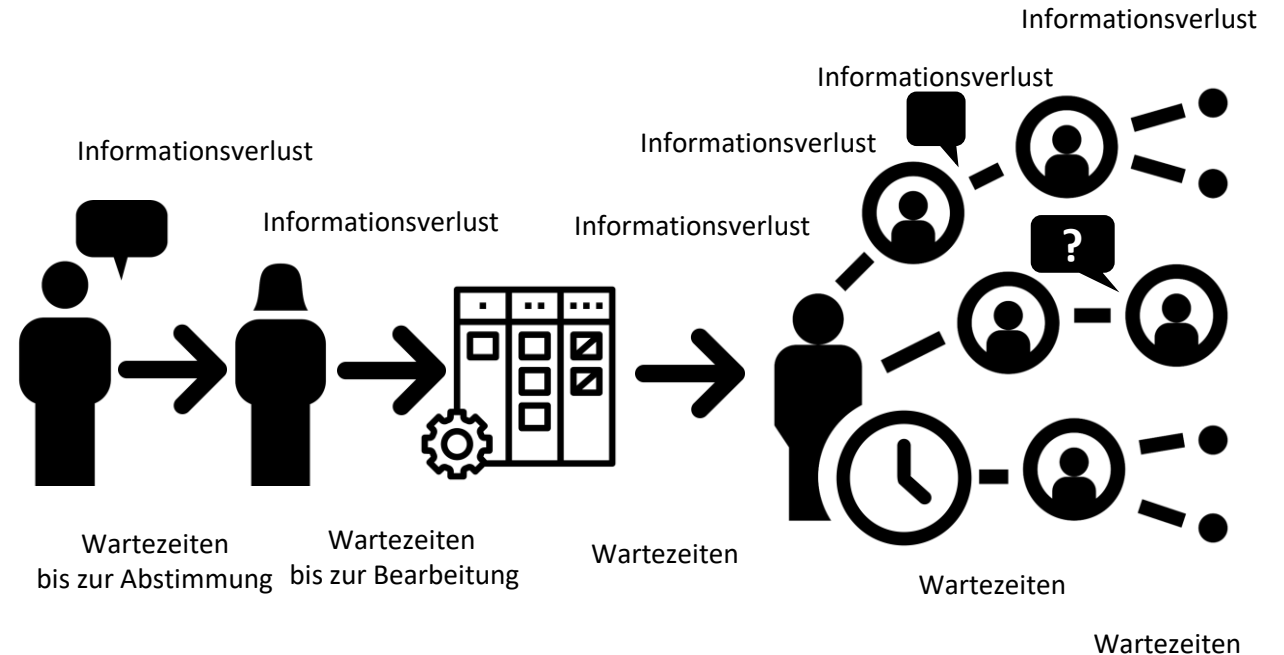
100-1.000

Z.B. Umgebungen erstellen, Server einrichten, Auschecken, Bauen, Testen, Administration, Storage verwalten, Netzwerk konfigurieren, Load Balancing, Sicherheitsmaßnahmen, noch mehr Testen, Freigaben...

# Übergaben

Bei jeder Übergabe

- Ist Kommunikation erforderlich
- Existiert die Gefahr von Wartezeiten
- Geht Information verloren



**Problem:** Ressourcen werden in unterschiedlichen Werteketten genutzt

# Vermeiden von Übergaben

## Maßnahmen

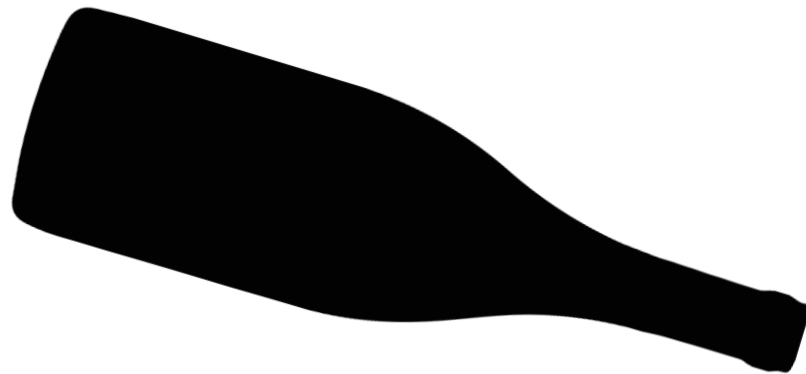
- Anzahl von Übergaben verringern
- Automatisierung
- Reorganisation der Teams

## Resultate

- Erhöhter Flow durch reduzierte Wartezeiten
- Zeit reduziert, in der kein Wert generiert wird

# Flaschenhalse identifizieren

- Es gibt in Werteketten genau **eine** Fließrichtung
- Es gibt genau **einen** Flaschenhals
- Optimierungen **vor** dem Flaschenhals verschlimmern den Rückstau
- Optimierungen **nach** dem Flaschenhals verschlimmern die Idle-Zeit



# Fünf fokussierende Schritte

- Den Flaschenhals finden
- Herausfinden, wie der Flaschenhals beseitigt werden kann
- Alles andere diesen Maßnahmen unterordnen
- Den Flaschenhals beheben
- Zurück zu Schritt 1 – jetzt wird es einen neuen Flaschenhals geben

# DevOps Muster

Lange Deployment-Laufzeiten folgen oft ähnlichen Mustern:

- Umgebungen
- Deployment
- Testen
- Architektur



# Verschwendung

Verschwendung (engl. waste, jap. Muda 無駄)

- Klassisch: Alles wofür ein Kunde nicht bereit ist zu zahlen, oder was er nicht benötigt [B1]
- Bei uns: Alles was Verzögerung verursachen kann [B2]

# Arten der Verschwendung

- Teilweise erledigte Arbeit
- Zusätzliche Prozesse
- Zusätzliche Features
- Aufgabenwechsel
- Warten
- Bewegung
- Fehler
- Nicht standardisierte oder manuelle Arbeit
- Heldentaten

# Zusammenfassung

Heute kennen gelernt:

- Übergaben vermeiden
- Flaschenhalse beseitigen
- Verschwendung vermeiden

# Quellenverzeichnis

- [B1] S. Shingo, A Study of the Toyota Productive System: From Industrial Engineering Viewpoint, Productivity Press, 1989
- [B2] M. Poppendieck, T. Poppendieck, Implementing Lean Software: From Concept to Cash, Addison-Wesley, 2007