

 IT-Tage  
2022

REMOTE  
KONFERENZ

12. - 15.12.2022

Marco Schulz



# DevOps 4 agile

Prozesse verstehen und richtig anwenden

#ittage

# Consultant



## **Elmar Dott**

(M. Schulz) studierte an der HS Merseburg Diplominformatik und twittert regelmäßig über alle möglichen technischen Themen. Seine Schwerpunkte sind hauptsächlich Build und Konfiguration Management, Software Architekturen und Release Management.

Seit über fünfzehn Jahren realisiert er in internationalen Projekten für namhafte Unternehmen umfangreiche Webapplikationen. Er ist freier Consultant / Trainer. Sein Wissen teilt er mit anderen Technikbegeisterten auf Konferenzen, wenn er nicht gerade wieder einmal an einem neuen Fachbeitrag schreibt.

<https://elmar-dott.com>

+ Consultant + Writer + Speaker + Trainer +

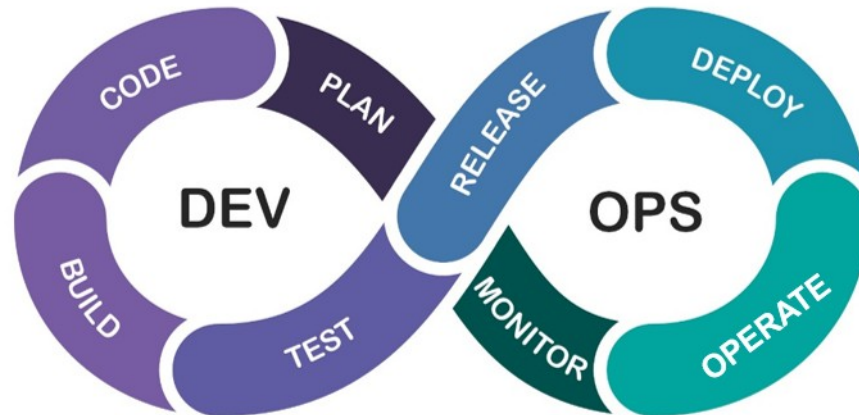
# Agenda

- 👤 Was ist DevOps?
- 👤 DevOps als Teil der agilen Kultur?
- 👤 Was sind die Voraussetzungen?
- 👤 Die Ausprägungen von DevOps.
- 👤 Ein Beispiel für Standardisierung.
- 👤 Der DevOps Werkzeugkasten.

# Was ist DevOps?

## Software-Entwicklungsprozess

**Entwicklung  
(DEVelop)**



**Betrieb  
(OPerationS)**

# DevOps als Agile Kultur

- alle Beteiligten verstehen sich als geschlossene Einheit
- nahtloser Informationsaustausch
- etablierten Standards folgen
- hohe Automatisierung in allen Bereichen

# Chancen & Risiken

- schnelles Time2Market
- höhere Kundenzufriedenheit
- bessere Produktqualität
- langfristig Reduktion der Entwicklungsaufwände
- Umdenken in den einzelnen Organisationseinheiten
- zu positive Annahme des Ausgangspunktes
- verlieren in technischen Details

# Paradoxon der Automatisierung



“A fool with a tool, is still a fool”

Gardy Booch

Prozessautomation reduziert das Risiko, dass Fehler gemacht werden. Aber hochkomplexe Prozesse sind naturgemäß nur sehr schwer zu automatisieren!

## Reproduzierbarkeit & Nachvollziehbarkeit

Reproduzieren bedeutet ein bereits erzielt Ergebnis unter gleichen Bedingungen beliebig oft erneut und ohne Abweichungen zu erhalten.

→ **Wiederholung**

Nachvollziehen heißt den Weg wie ein Ergebnis erzielt wurde vollständig zu erfassen, um diesen bei Bedarf anpassen zu können.

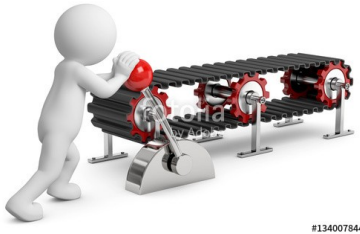
→ **Verständnis**



# Voraussetzungen

- Unternehmensprozesse sollten möglichst vereinheitlicht und standardisiert sein (KISS Prinzip)
- Vollständige und möglichst automatisierte Teststrategie
- Infrastruktursysteme dürfen nicht manuell verändert werden

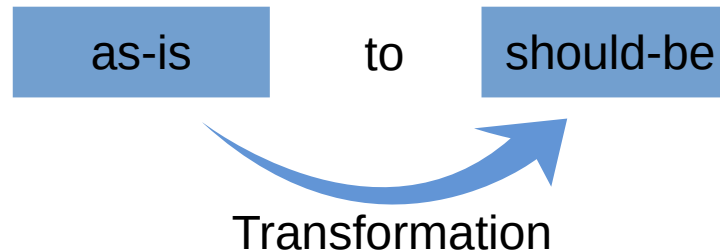
# Der Prozess



EN ISO 900: Ein **Prozess** ist ein Satz von in Wechselbeziehung stehenden Tätigkeiten, der Eingaben in Ergebnisse umwandelt.

- autonom
- interaktiv

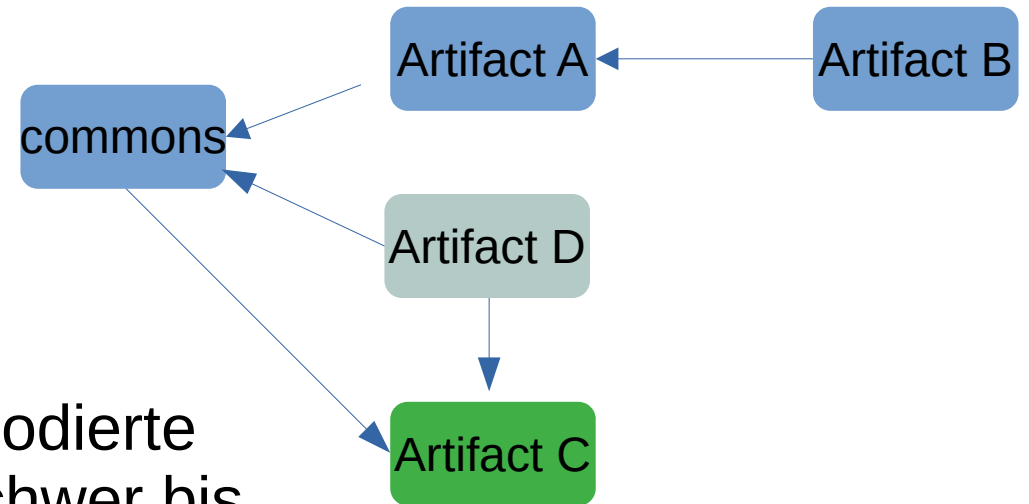
Optimierung:



# Architekturprobleme

Wichtige Eigenschaften einer Architektur:

- Einfach zu verstehen
- Leicht zu benutzen
- Fehlertolerant



Ungünstig entworfene oder erodierte Architekturen sind nur sehr schwer bis unmöglich zu automatisieren.

# Interaktionen

**Choreographie:** beschreibt einzelne Operationen, aber nicht die Nachrichtenfolge (Ablauf). Es behandelt die etablierte Kommunikation zwischen zwei Teilnehmern.

**Orchestration:** beschreibt die Reihenfolge und Bedingungen der Aufrufe beliebiger Teilprozesse.

**Konversation:** beschreibt die Abfolgen zwischen Prozessen. Es wird die gesamte zulässige Kommunikation (Vollständigkeit) zwischen zwei Teilnehmern beschrieben.

# Begrifflichkeiten

**Release:** (dt. : loslassen) bezeichnet also einen definierten Zustand eines Artefaktes, dessen Entwicklungszyklus abgeschlossen wurde, so dass es zur Verwendung freigegeben ist.

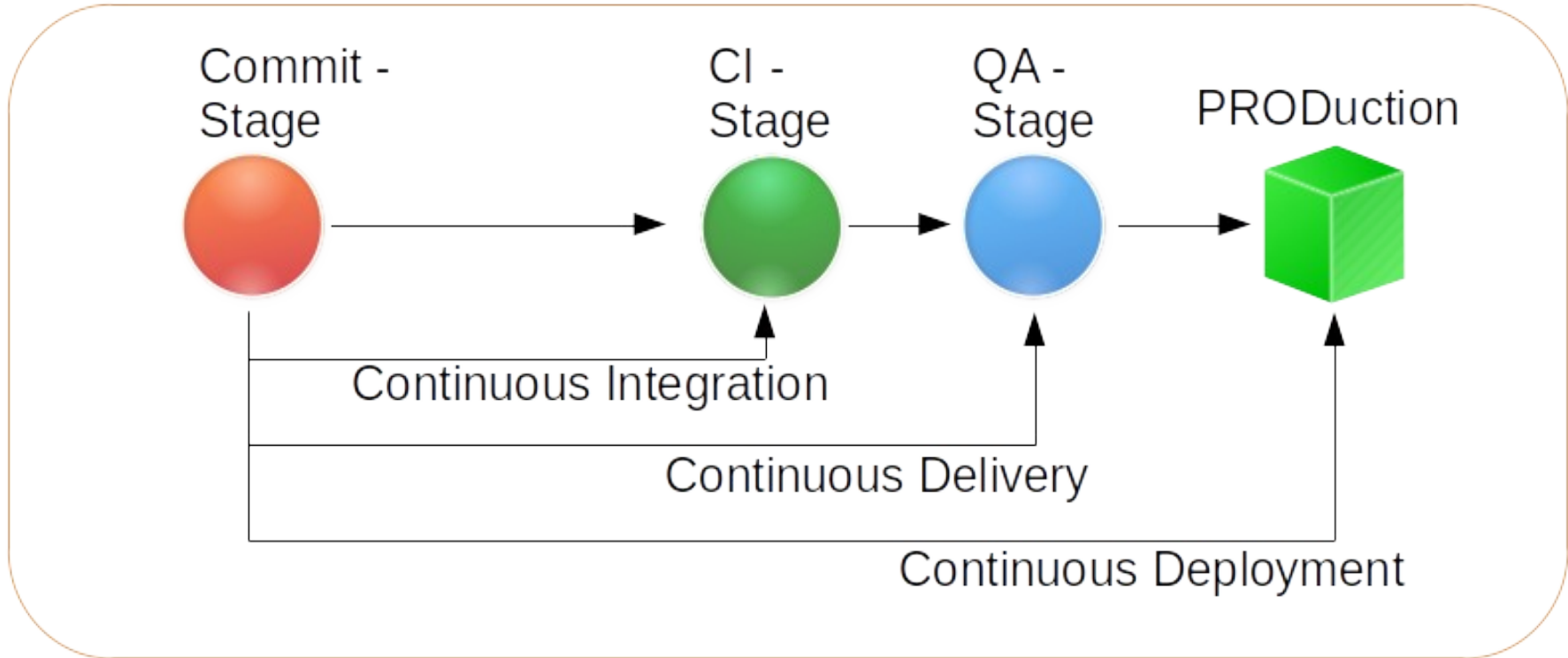
**Deliver:** (dt. : Auslieferung) bezeichnet die Bereitstellung der Artefakte zur weiteren Verwendung, z. B. durch die Installation auf einem Repository Server.

**Deploy:** (dt. : Einsatz) meint die Installation einer Software in die Zielumgebung, vornehmlich nach PRODUKTION.

# Prozesskette



# Automatisierungsstufen



# Projektplan

Weeks	01	02	03	04	05	06	07	08	09	Version
Sprint 1	D	T								1.1.0
Sprint 2		D	T							1.2.0
Sprint 3			D	T						1.3.0
<b>Sprint 4</b>				K/T						<b>1.4.0 Prod.Cand.</b>
Sprint 5				D	T					1.5.0
Sprint 6					D	T				1.6.0

D = Develop / T= Test / K = Corrections



# Fehlerklassen

Fehlerklassen die eine Verwendung des Releases verhindern können:

- Einfache Fehler (GUI/ Layout/ Beschriftung)
- Mittlere Fehler („einfache“ Logikfehler)
- **Schwere Fehler (Abnahme verhindernd)**

~~Release Candidate~~ → **Production Candidate**

# Der DevOps Werkzeugkasten

Es gibt verschiedene Werkzeuge, die für DevOps Prozesse in Frage kommen. **Dabei ist erst der Prozess zu definieren, um dazu das passende Werkzeug auszuwählen.**

- Gibt es bereits im Unternehmen Kompetenzen zu einem Werkzeug / einer Technologie?
- Wie zukunftssicher ist die Investition?
- Ist das Budget im Rahmen?
- Hat das Werkzeug einen wirklichen Mehrwert für die tägliche Arbeit



**Nie mehr als 3 neue Technologien gleichzeitig einführen!**

# Credentials



Homepage : <https://elmar-dott.com>  
GitHub : <https://github.com/ElmarDott>  
AnchorFM : <https://anchor.fm/elmar-dott>  
Twitter : <https://twitter.com/ElmarDott>  
Speaker Deck : <https://speakerdeck.com/elmardott>

Lbry : <https://lbry.tv/@elmar.dott:8>  
BitChute : <https://www.bitchute.com/channel/3IyCzKdX8Ip0/>



**Danke / thank you / Gracias**

# Referenzen

- [1] Marco Schulz, 2021, Continuous Integration mit Jenkins, Rheinwerk, ISBN: ISBN 978-3-8362-7834-8  
<https://www.rheinwerk-verlag.de/continuous-integration-mit-jenkins/>
- [2] Robert C. Martin, 2018, Clean Architecture, Person, ISBN 0-13-449416-4
- [3] US Ministry of Defense, 2018, DIB Guide: Detecting Agile BS
- [4] Marco Schulz, javaAktuell, 1.2023, Prozesslandschaften,  
<https://elmar-dott.com/publications/prozesslandschaften/>



# Seminare & Trainings

<https://elmar-dott.com/courses/de/>





**BITCHUTE**

