



Marco Schulz

Continuous Stupidities – DevOps Mythen



2024

DEVOPSDAYS

Seminare



1. Web Application Security
2. DevOps Praxistraining (mit Spring Boot)
3. DevOps Kultur & Werkzeuge
4. DevOps Praktiken und Automatisierung
5. Testing with JUnit & Mockito
6. Softwarewartung - Projekte zukunftssicher gestalten
7. Schnelleinstieg in Apache Maven
8. Git für Entwickler
9. Docker für (Web) Entwickler
10. Docker Essentials
11. Linux für Anwender
12. Linux Technologieüberblick für Führungskräfte
13. Einführung in die Datenbankentwicklung
14. Grundlagen des Datenbank Entwurfs
15. Grundlagen der OOP

Vorstellungsrunde



Elmar Dott

Marco Schulz, im Internet auch unter seinem Pseudonym Elmar Dott bekannt, realisiert seit über knapp 20 Jahren als freier Berater in internationalen Projekten große Java EE Web-Applikationen.

Seine Schwerpunkte sind DevOps, Konfigurationsmanagement, Software-Architekturen & Release Management. Als Trainer teilt er sein Wissen in Schulungen und spricht auch regelmäßig auf Konferenzen über aktuelle Themen. Er ist Autor des Buches „CI mit Jenkins“ und hat zudem unzählige Fachartikel publiziert.

<https://elmar-dott.com>

+ Consultant + Writer + Speaker + Trainer +

Inhalte

- Was ist DevOps?
- Ein paar Worte zu Testautomatisierung
- Begriffe und Definitionen
- Continuous was?
- Der Continuous Workflow
- Die Bedeutung von Conway's Gesetz
- Semantic Versioning
- Wiederverwendung – Der Umgang mit Ressourcen



Die vollständigen Folien zum Vortrag:

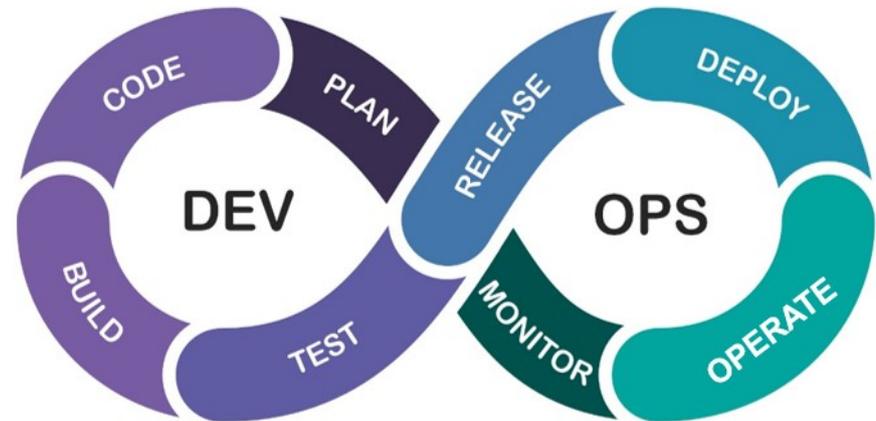
https://elmar-dott.com/event/devops-cloud-tage-2024-remote/?mtm_campaign=SpeakerDesk&mtm_kwd=de

Was ist DevOps?

DEVELOpERations

- Agile Kultur
- Betrieb (Ops) ist in den Entwicklungsprozess integriert
- nahtloser Informationsaustausch
- etablierten Standards folgen
- Automatisierung steht über manuellen Prozessen

Software Development Lifecycle



Das Automatisierungs Paradoxon

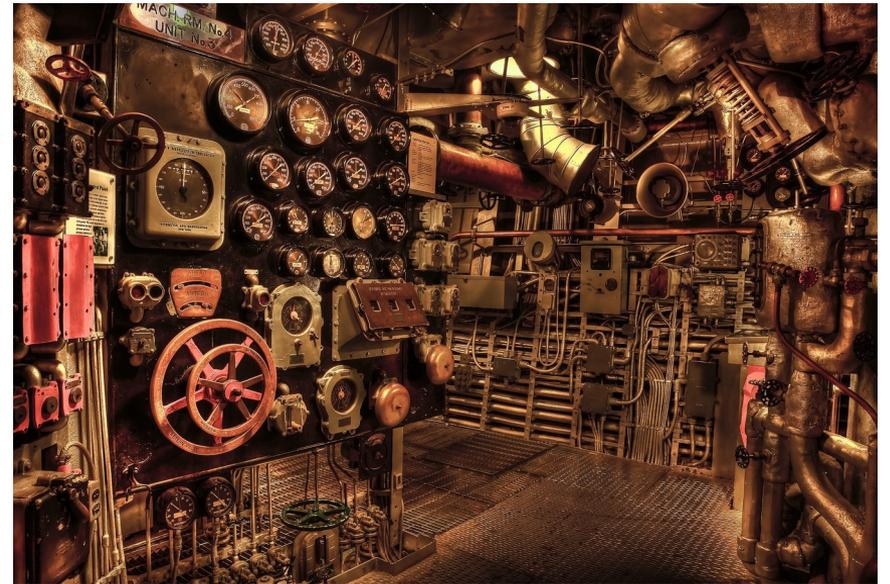


“A fool with a tool, is still a fool”
Gardy Booch

Prozessautomatisierung verringert das Risiko von Fehlern, **hochkomplexe Prozesse sind jedoch oft schwer zu automatisieren.**

Testautomatisierung

Manuelle Tests statt automatisierter Tests während des SDLC indizieren, dass eine Organisation Lichtjahre von einem continuous Paradigma entfernt ist.



Definitionen

Release: (package) – wenn wir den Implementierungszyklus abgeschlossen haben, packen wir das Artefakt

(1) to give up in favor of another

(2) to make available to the public

Deploy: (storage) – das Artefakt wird für die Wiederverwendung gespeichert

(1) to spread out, utilize, or arrange for a deliberate purpose

Deliver: (install) – die Anwendung zur Verwendung in eine Umgebung installieren

(1) to take and hand over to or leave for another

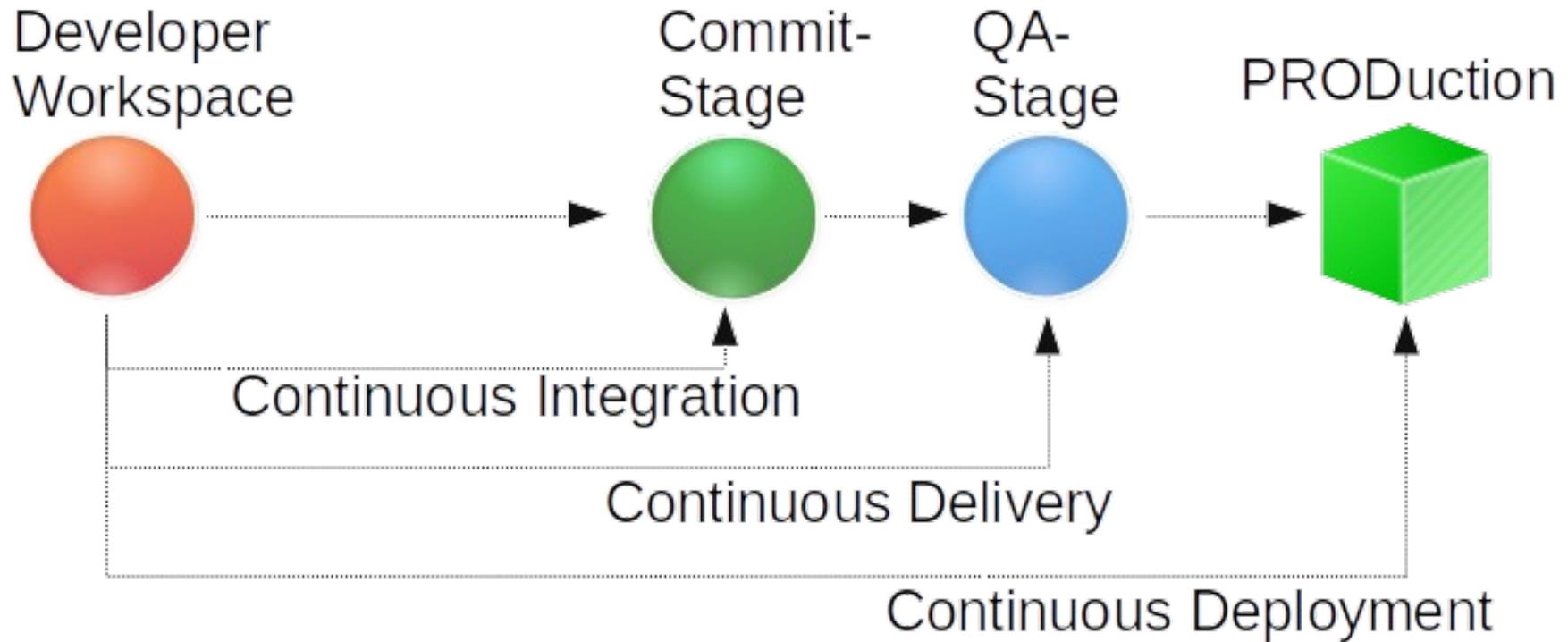
<https://www.merriam-webster.com>

Continuous was?

We use the term continuous in this book, but the usage is technically incorrect. Continuous implies that something kicks off and never stops. This suggests that the process is constantly integrating, which is not the case in even the most intense CI environment. So, what we are describing in this book is more like **continual** integration.

Paul M. Duvall

Paradigmen



Paradigmen

- **Continuous Integration:** Hält den Quellcode immer in einem kompilierbaren Zustand.
- **Continuous Delivery:** Hält die Software immer in einem freigabefähigen (release) Zustand.
- **Continuous Deployment:** Hält die Software immer in einem installierbaren Zustand.

Der Continuous Workflow

Schnelles Feedback (agil): Schnell scheitern und die Informationen zur richtigen Zeit an die richtigen Personen weitergeben

→ Bei **Informationsüberflutung** ist immer jeder an jeder Benachrichtigung beteiligt – das führt dazu, dass jeder alle Benachrichtigungen ignoriert

Continuous Integration

- die Commit Stage ist **kein** Qualitätsziel
- nicht jeder Commit muss zwangsläufig ein Build auslösen
- nightly build bedeutet heutzutage nightly deploy
- Release Candidate sollte besser als Production Candidate bezeichnet werden

Conway's Law

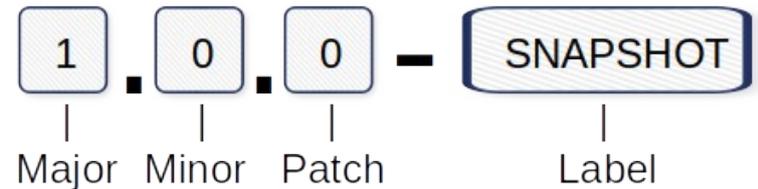
“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization’s communication structure.”

CONWAY, MELVIN E.

Folgen Sie den Standard und transformieren (vereinfachen) Sie vorhandene Prozesse, anstatt etablierte Lösungen so zu modifizieren das sie an eigene Bedürfnisse angepasst sind.

Semantic Versioning

Angesichts einer Versionsnummer [MAJOR . MINOR . PATCH], erhöhen Sie wie folgt:



1. **MAJOR** - wenn Sie inkompatible API-Änderungen vornehmen,
2. **MINOR** - wenn Sie Funktionalität rückwärtskompatibel hinzufügen und
3. **PATCH** - wenn Sie abwärtskompatible Fehlerbehebungen vornehmen.

Zusätzliche Bezeichnungen für Vorabversions- und Build-Metadaten sind als Erweiterungen des MAJOR.MINOR.PATCH-LABEL verfügbar.

Wiederverwendung

Alles von Grund auf bauen zu können, bedeutet nicht, dass wir immer alles von Grund auf bauen!

- Release Artefakte werden für die Testautomatisierung wiederverwendet
- Pulled container aus dem DockerHub sollten gecached werden
- Eigene Images sollten in einem Repository Manager abgelegt werden

Rollback

Wie verwalten Sie Bereitstellungsfehler oder Anwendungsfehler?

→ Blue / Green Deployment
Berücksichtigt keine Daten-
migration zwischen Tabellen
und Schemata Änderungen.



Referenzen

[01] Marco Schulz, 2021, Continuous Integration mit Jenkins, Rheinwerk, ISBN: ISBN 978-3-8362-7834-8
<https://elmar-dott.com/buecher/ci-jenkins/>

[02] J. Humble & D. Farley, Continuous Delivery, 2011, Addison Wesley, ISBN: 0-321-60191-2

[03] Paul M. Duvall, Continuous Integration, 2007, Addison Wesley, ISBN: 0-32133638-0

[04] Marco Schulz, javaAktuell, 1.2023, Prozesslandschaften,
<https://elmar-dott.com/publications/prozesslandschaften/>

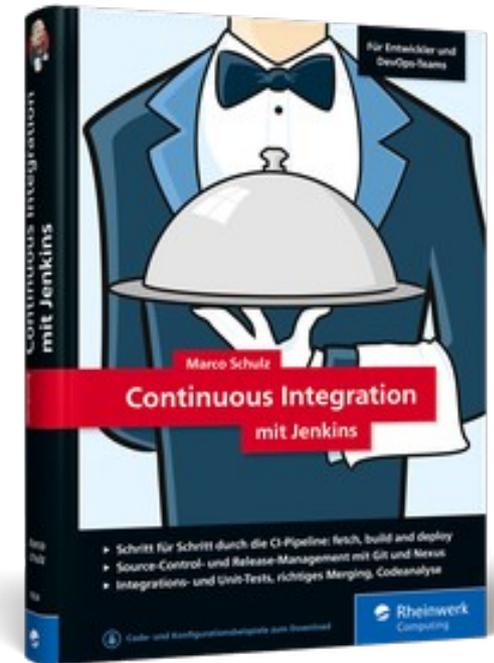
[05] CONWAY, MELVIN E. "HOW DO COMMITTEES INVENT?" 1968, DATAMATION, VOL. 14, NUM. 4, PP. 28-31

[06] Robert C. Martin, 2018, Clean Architecture, Person, ISBN 0-13-449416-4

[07] <https://elmar-dott.com/articles/conways-law/>

[08] <https://semver.org>

[09] <https://dzone.com/articles/version-number-anti-pattern>





Credentials

Software, Consulting & Training



Homepage : <https://elmar-dott.com>

E-Mail : elmar.dott@gmail.com



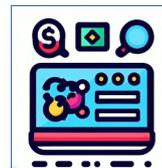
VPS Configuration & Administration



Marketing & eCommerce



Application Development



Web Development



Coaching & Trainings