Marco Schulz

Continuous Stupidities - Myths of DevOps

NOV 20-22

JCON WORLD 2023 ONLINE

JAVAPRO # JCON2023 www.jcon.one

Workshops & Training's

https://elmar-dott.com/courses/

Consultant

Elmar Dott

Marco Schulz, also known on the Internet under his pseudonym Elmar Dott, has been implementing large web applications as a freelance consultant in international projects for over 20 years. His focus is on DevOps, configuration management, software architectures & release management. As a trainer, he shares his knowledge in training courses and also speaks regularly on current topics at conferences. You can find his homepage at:

https://elmar-dott.com

+ Consultant + Writer + Speaker + Trainer +

Agenda

- What is DevOps?
- Some words about test automation
- A mismatch of definitions.
- Continuous what?
- Paradigms
- Managing the Continuous workflow
- Ignoring Conway’s law.
- Semantic Versioning
- Reuse – Managing Resources
- Handle the worst case.

Find the full slides including the record at:

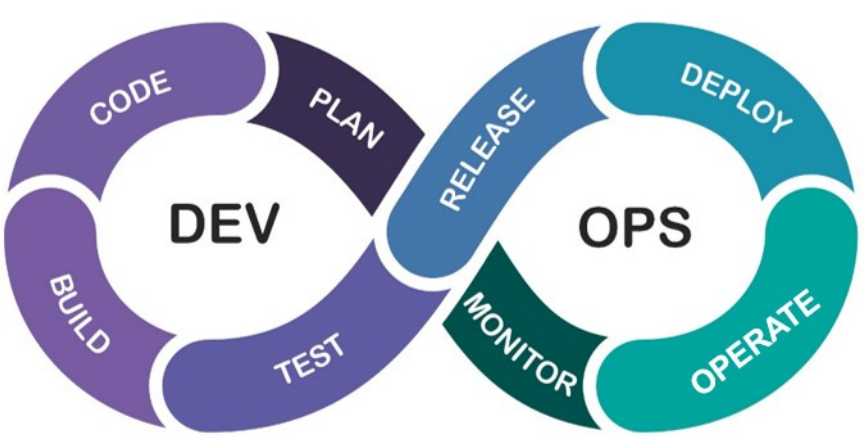
https://elmar-dott.com/event/jcon-world-2023/?mtm_campaign=SpeakerDesk&mtm_kwd=en

What is DevOps?

DEVelop**OP**eration**S**

- agile culture
- operations is involved in the development process
- seamless information flow
- follow established standards
- automation instead manual process

Software Development Lifecycle



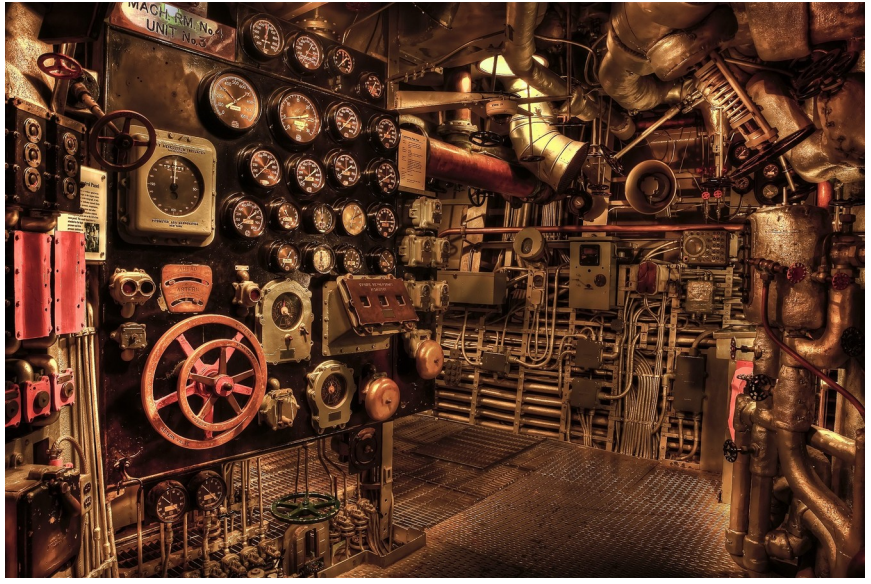
The automation paradox

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

Process automation reduce the risk of failures, but **high complex processes are often hard to automate**.

Test Automation

Manual test instead of automated test during the SDLC indicates an organization is light years far away of any continuous paradigm.



Definitions

Release: (package) – when we finish the implementation cycle we pack the artifact

(1) to give up in favor of another

(2) to make available to the public

Deploy: (storage) – the artifact have to be stored for utilize in previous stages

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

Deliver: (install) – bring the application into an environment

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

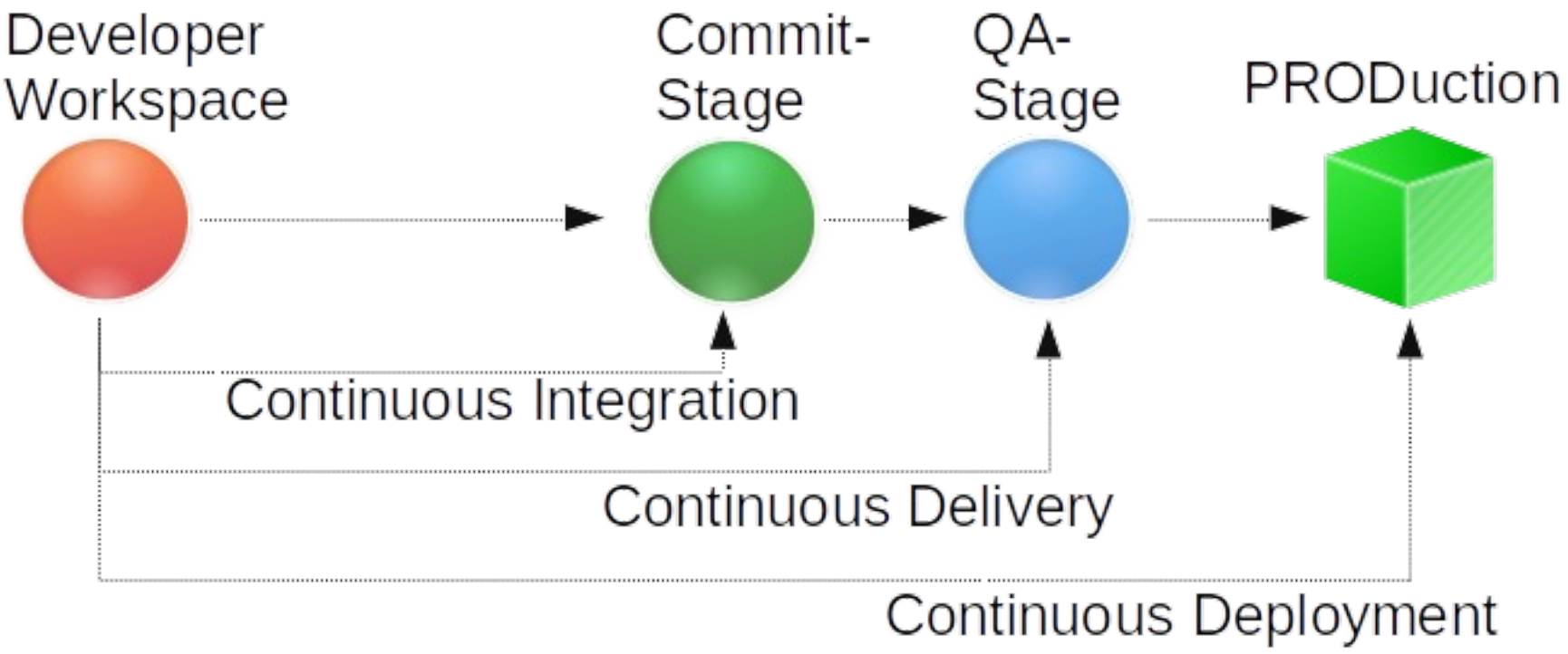
https://www.merriam-webster.com

Continuous what?

We use the term continuous in this book, but the usage is technicality incorrect. Continuous implies that something kicks off and never stops. This suggest 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

Paradigms



Paradigms

- **Continuous Integration**: keeps the source code always in a compilable state.
- **Continuous Delivery**: keeps the software always in a releasable state.
- **Continuous Deployment**: keeps the software always in a installable state.

Managing the continuous workflow

Rapid Feedback (agile): **Fail fast** and deliver the information to the right persons at the right time

→ information overload is when everybody is always involved in every notification – this causes everyone will ignore all notifications

Continuous Integration

- the commit stage is **not** a quality gate
- not every commit need to trigger a build
- nightly build today means nightly deploy
- Release Candidate should called Production Candidate

Conway’s Law

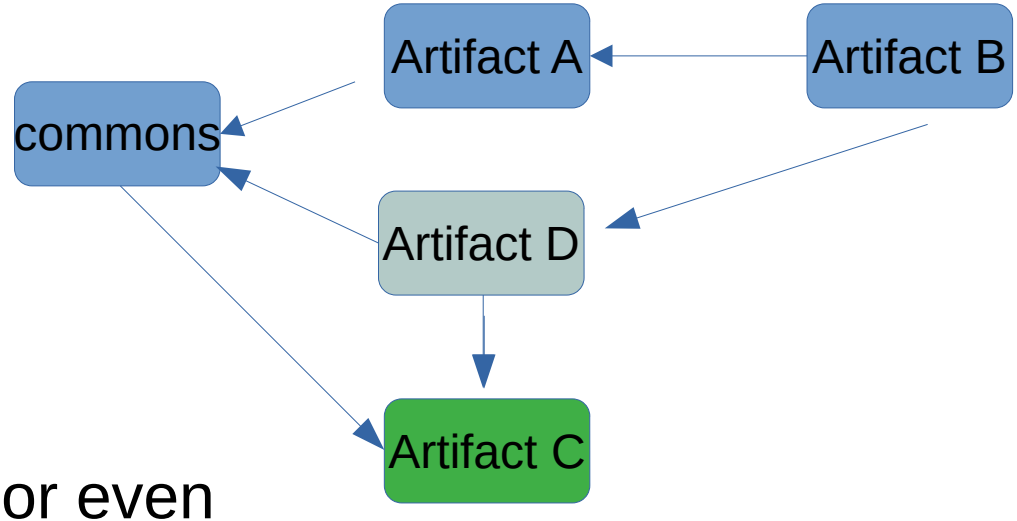
> “Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization’s communication structure.”
>
> CONWAY, MELVIN E.

Follow the standard and transform (simplify) your processes instead of modify or customize established solutions to fit to your needs.

Software Architecture

Important values of an architecture:

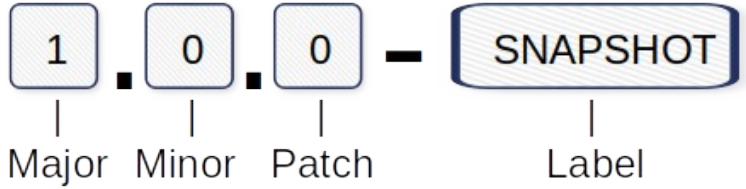
- Easy to understand
- Easy to use
- Fault tolerant



Poorly designed or eroded architectures are very difficult or even impossible to automate.

Semantic Versioning

Given a version number [MAJOR . MINOR . PATCH], increment the:



1. **MAJOR** - when you make incompatible API changes,
2. **MINOR** - when you add functionality in a backwards-compatible manner, and
3. **PATCH** - when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Reuse – Managing Resources

To be able to build everything from the scratch don’t mean we build **always** everything from the scratch!

- Released artifacts will reused for test automation
- Pulled containers from DockerHub have to be cached
- Created Images will be stored in a Repository Manager

Rollback

How to manage deployment failures or application errors?

→ Blue / Green Deployment does not manage data integration between schemata / table changes.

References

[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

Web Development

Marketing & eCommerce

Coaching & Trainings

Application Development