



Elmar Dott

Estupidencias
Continuas:
Mitos de
DevOps



Seminarios



1. Web Application Security
2. DevOps Praxistraining (Java, Spring Boot)
3. DevOps Culture & Tools
4. DevOps Practice and Automation
5. Testing with JUnit & Mockito
6. Software maintenance - future-proofing projects
7. Build Management with Apache Maven
8. Git for Developers
9. Docker for (Web) Developer
10. Docker Essentials
11. Linux Essentials
12. Linux: Decisionmaking for Managers
13. Database Essentials
14. Database Development & Design
15. Introduction into OOP

Consultant



Elmar Dott

Marco Schulz, también conocido en Internet con el seudónimo de Elmar Dott, lleva más de 20 años implementando grandes aplicaciones web como consultor autónomo en proyectos internacionales. Su trabajo se centra en DevOps, gestión de la configuración, arquitecturas de software y gestión de versiones. Como formador, comparte sus conocimientos en cursos de formación y también habla regularmente sobre temas de actualidad en conferencias. Puede encontrar su página web en:

<https://elmar-dott.com>

+ Consultant + Writer + Speaker + Trainer +

Agenda

- Qué es la DevOps?
- Un desajuste de definiciones.
- Continuous qué?
- Reutilización - Gestión de recursos.
- Gestión del flujo de información.
- Paradigmas!
- Estupidencs Continues



Encontrará las diapositivas completas, incluido el disco, en:

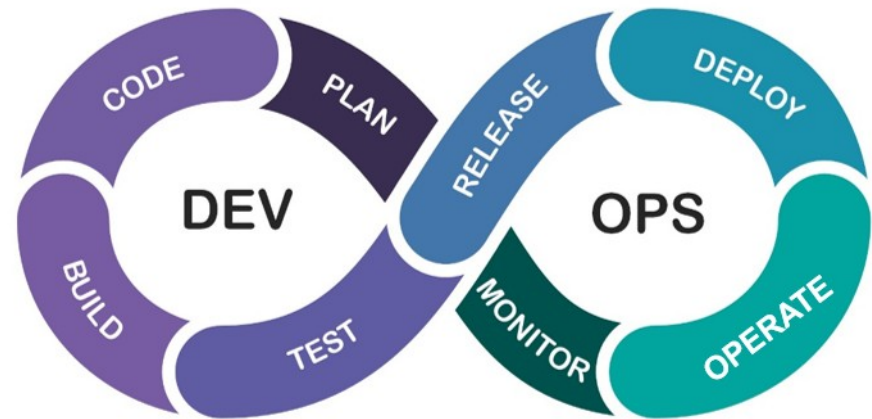
https://elmar-dott.com/event/devops-days-medellin-2024/?mtm_campaign=SpeakerDesk&mtm_kwd=es

Qué es la DevOps?

DEVELOP^{er}ATIONS

- cultura ágil
- las operaciones participan en el proceso de desarrollo
- flujo de información fluido
- sigue normas establecidas
- automatización en lugar de procesos manuales

Software Development Lifecycle



La paradoja de la automatización



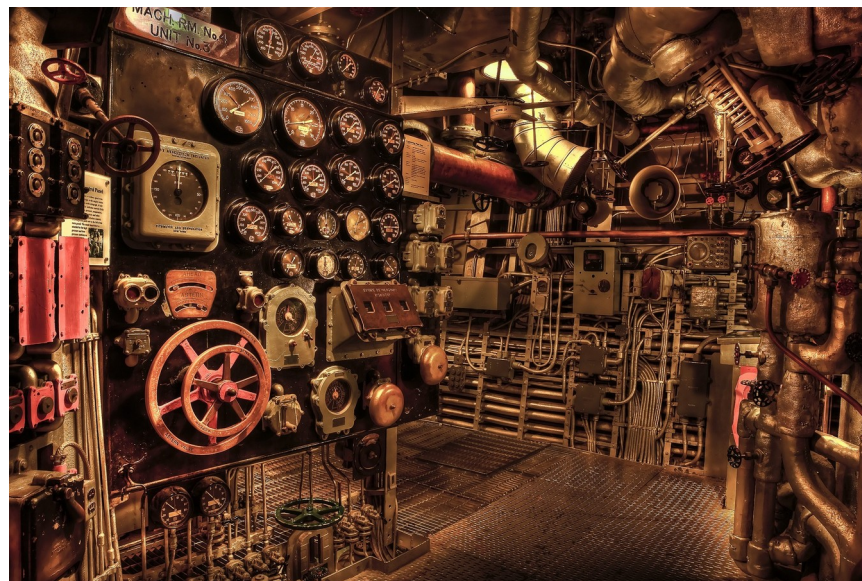
“Un tonto con una herramienta,
sigue siendo un tonto”

Gardy Booch

La automatización de procesos reduce el riesgo de fallos, pero **los procesos de gran complejidad suelen ser difíciles de automatizar.**

Automatización de pruebas

La prueba manual en lugar de la prueba automatizada durante el SDLC indica que una organización está a años luz de cualquier paradigma continuo.



Definiciones

Release: (paquete) – cuando terminamos el ciclo de implementación empaquetamos el artefacto

- (1) to give up in favor of another
- (2) to make available to the public

Deploy: (almacenar) – el artefacto debe almacenarse para su utilización en etapas anteriores

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

Deliver: (instale) – introducir la aplicación en un entorno

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

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

Continuous qué?

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

Reutilización - Gestión de recursos

Poder construir todo desde cero no significa que **siempre** construyamos todo desde cero.

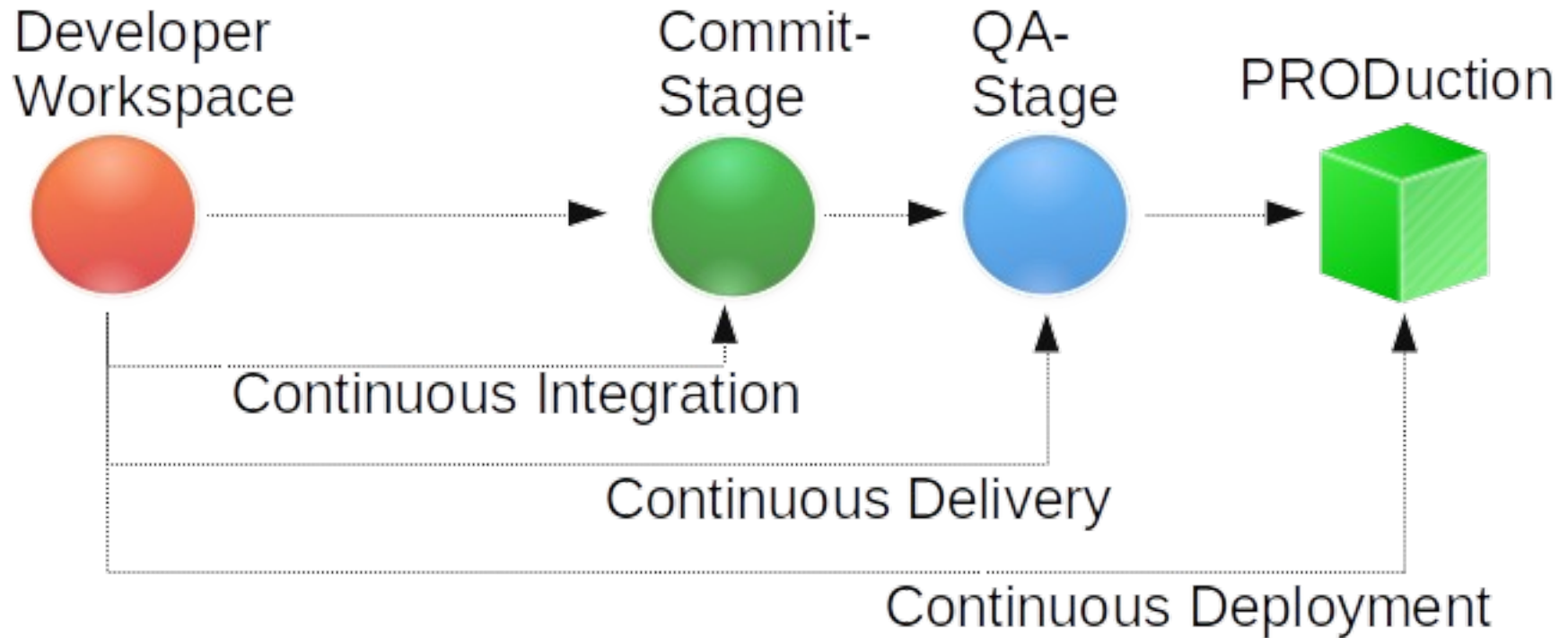
- Los artefactos liberados se reutilizarán para la automatización de pruebas
- Los contenedores extraídos de DockerHub deben almacenarse en caché
- Las Imágenes creadas se almacenarán en un Gestor de Repositorios

Gestión del flujo de información

Retroalimentación rápida (ágil): Fallar rápido y entregar la información a las personas adecuadas en el momento adecuado.+

→ sobrecarga de información es cuando todo el mundo está siempre involucrado en cada notificación - esto hace que todo el mundo ignore todas las notificaciones

Paradigms



Paradigms

- **Continuous Integration:** mantiene el código fuente siempre en estado **compilable**.
- **Continuous Delivery:** mantiene el software siempre en un estado **releaseable**.
- **Continuous Deployment:** mantiene el software siempre en estado **installable**.

Continuous Stupidities

- the commit stage **no es** una puerta de calidad
- no todas las commits tienen que desencadenar una compilación
- nightly build hoy significa nightly deploy
- Release Candidate debe llamarse Production Candidate

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://dzone.com/articles/version-number-anti-pattern>





Credentials

Software, Consulting & Training



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

E-Mail : elmar.dott@gmail.com



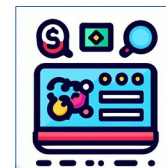
VPS Configuration & Administration



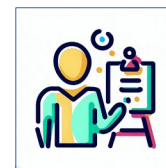
Marketing & eCommerce



Application Development



Web Development



Coaching & Trainings