

# DevOps

## **Aufgabe 1: Von der Idee zum Produkt**

1. Zufällige 4er Gruppen
2. Plakat erstellen über die Schritte welche nötig sind um Programmcode von der Idee bis zum Kunde zu bringen.
3. Präsentieren im Plenum

-  "Gruppenarbeit"
-  **30 Minuten**
-  Plakat
-  Vorwissen abfragen

# Development

## Software Entwicklung

# Projektmanagement

- Anforderungsanalyse
- Arbeitspakete
- Prozesssteuerung

## Oder auch

- 🧠 Den Kunden verstehen wollen
- 📌 Post-its voll malen
- 🧑‍🔧 Delegieren!



# Programmieren

- Neue Software erschaffen
- Bestehende Software verbessern/erweitern

## Oder auch

- 😄 Spass haben
- 😞 Kreativ sein!
- 🤔 Coding Themes ausprobieren

```
! release-please.yml x ! deploy.yml x index.html x ! docker-compose.yml x
1  services:
2    devcontainer:
3      build:
4        context: .
5        dockerfile: .devcontainer/Dockerfile
6      # Loop to not get down
7      command: /bin/sh -c "while sleep 1000; do :; done"
8      # start nginx
9      # command: /usr/sbin/nginx -g "daemon off;"
10     working_dir: /workspace
11     container_name: devcontainer
12     ports:
13       - "3000:3000"
14     # INFO auskommentieren, sobald "local" Ordner existiert
15     # env_file: ./local/.env
16     volumes:
17       # sync workspace
18       - ./workspace:cached
19
20       # sync nginx files
21       - ./nginx/src:/usr/share/nginx/html
22       - ./nginx/default:/etc/nginx/sites-available/default
23
24     # INFO auskommentieren, sobald "local" Ordner existiert
25     # - ./local/aws:/root/.aws
26     # - ./local/ssh/id_rsa.pem:/root/.ssh/id_rsa.pem
27   production:
28     build:
29       context: ./nginx
30       dockerfile: Dockerfile
31     ports:
32       - "3001:3000"
```

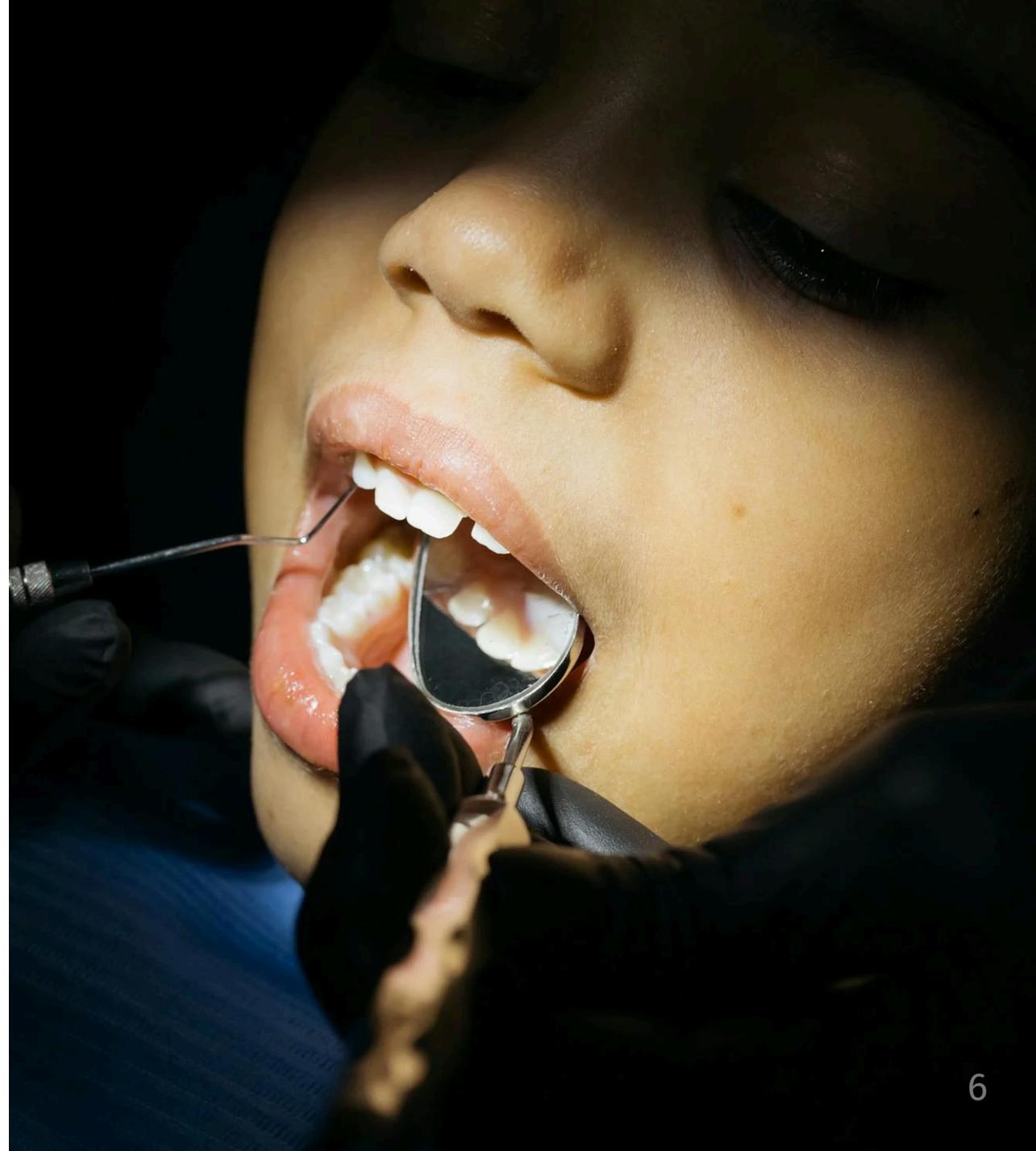
NORMAL main ! docker-compose.yml > services devcontainer gj 34% 11:1 21:21  
bbzbl-modul-324 bbzbl-modul-324-nginx

# Qualitätsmanagement

- Automatische Tests
- Nomenklatur (Formatierung)
- Codequalität garantieren (Linting)

## Oder auch

- 🦷 auf den Zahn fühlen
- 🚒 pingelig sein



# Ausliefern

- Software bauen
- Images erstellen (Containerisieren)
- In Betrieb nehmen

## Oder auch

- 🌱 Ein Samen setzen
- 🍀 Software ist im stetigen Wandel



# Operations

## IT Betrieb

# Infrastruktur

- Hardware
- Virtuelle Maschinen
- Cloud



# Service

- Picketdienst
- Monitoring
- Backup / Failure recovery



# Apps betreiben

## Konventionell

- Artefakte (.jar / .war / Quellcode)
- Manuelle Konfiguration der Servers

## Container

- Docker / Images
- Docker Compose / Swarm
- Kubernetes / OpenShift



# Was ist nun *DevOps*?

# DevOps

Laut [Wikipedia](#) existiert **keine universelle Definition** vom Begriff *DevOps*.



Der Begriff ist **nicht geschützt!**

Es handelt sich um ein sogenanntes **Buzzword!**

**Wieso haben wir dann ein  
Modul dafür?**

# DevOps / *Prinzipien*

DevOps verbindet den **IT-Betrieb** mit der **Software Entwicklung** durch

---



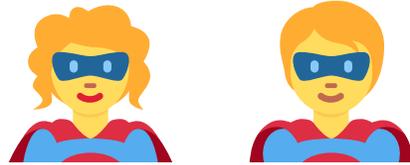
Geteilte Verantwortung



Arbeitsprozess Automatisierung



Schelles Feedback



# *Jeder kann und kennt alles*

Kein Gärtchendenken

Kein Verantwortung-Abschieben

**Kein Problem Ferien zu machen**



# ***Mr. Robot* hilft dabei!**

Keine repetitiven Tasks

Konventionen werden eingehalten

Flüchtigkeitsfehler werden vermieden



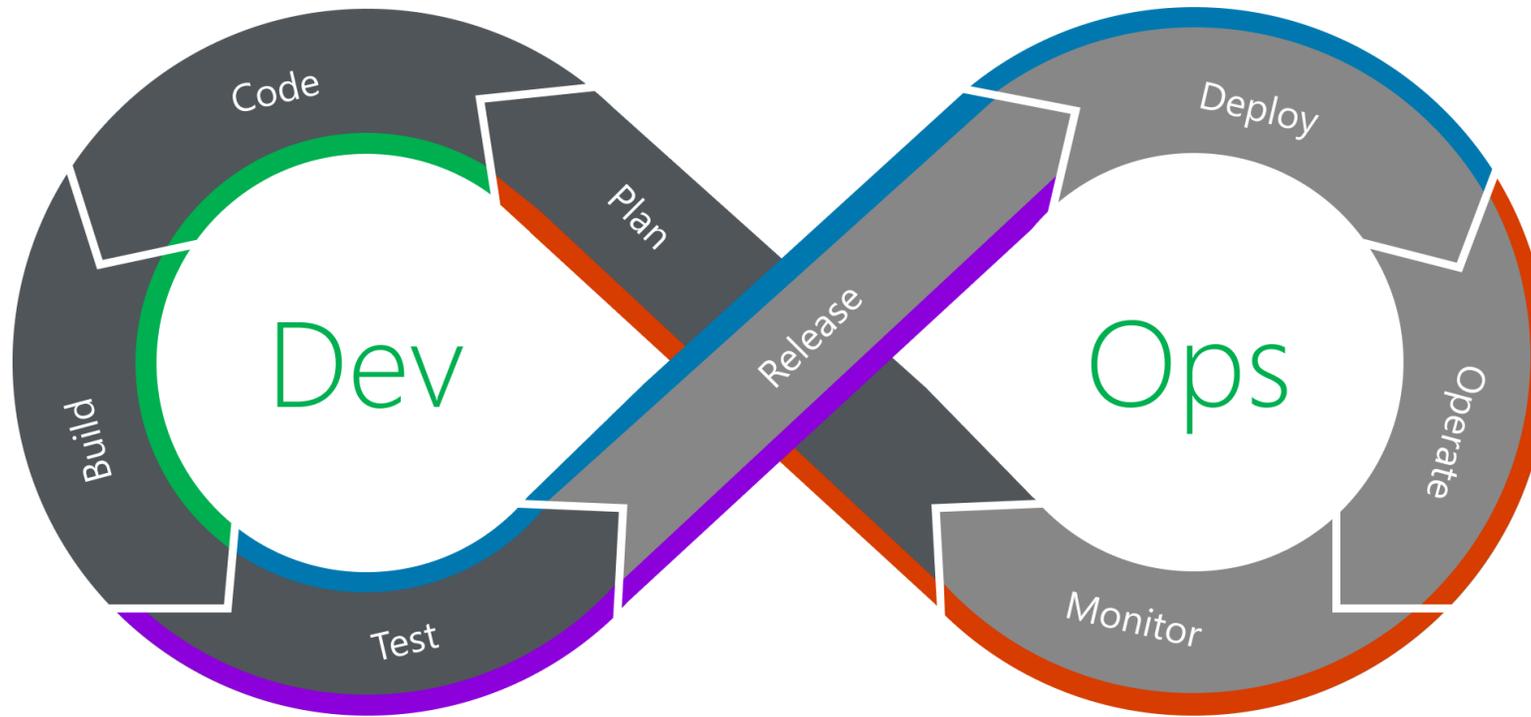
# ***Kunde hilft dabei!***

Schnelles Feedback

Hohe Zufriedenheit

Hohes Vertrauen

## Communication, Collaboration and Security



■ Continuous Integration (CI)

■ Continuous Deployment (CD)

■ Continuous Delivery (CD)

■ Continuous Feedback (CF)

# DevOps ist ein Mindset

Ausfallsicherheit

Risikominimierung

Kundenzufriedenheit

Mitarbeiterzufriedenheit