

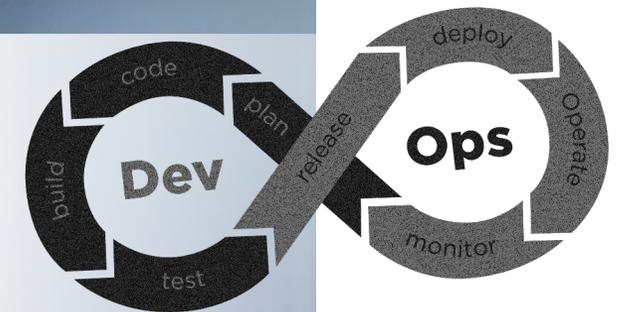


20
MIN

CLOUD-NATIVE

Unit:
DevOps

(1) Was ist das? DevOps Prinzipien



Urheberrechtshinweise

Diese Folien werden zum Zwecke einer praktikablen und pragmatischen Nutzbarkeit im Rahmen der **CCo 1.0 Lizenz** bereitgestellt.

Sie dürfen die Inhalte also kopieren, verändern, verbreiten, mit eigenen Inhalten mixen, auch zu kommerziellen Zwecken, und ohne um weitere Erlaubnis bitten zu müssen.

Eine Nennung des Autors ist nicht erforderlich (aber natürlich gern gesehen, wenn problemlos möglich).

Diese Folien sind insb. für die Lehre an Hochschulen konzipiert und machen daher vom **§51 UrhG (Zitate)** Gebrauch.

Die CCo Lizenz überträgt sich nicht auf zitierte Quellen. Hier sind bei der Nutzung natürlich die Bedingungen der entsprechenden Quellen zu beachten.

Die Quellenangaben finden sich auf den entsprechenden Folien.



KAPITEL 3 + 6

Cloud Computing und Cloud-native



3 DevOps

3.1 Prinzipien des Flow

Arbeit sichtbar machen, Work in Progress beschränken, Flaschenhalse minimieren

3.2 Prinzipien des Feedbacks

Probleme früh erkennen, Probleme sofort lösen, Probleme professionell verantworten

3.3 DevOps-geeignete Architekturen

Randbedingungen für die Entwicklung, Nutzung von Orchesterungsplattformen, Randbedingungen für den Betrieb

6 Deployment-Pipelines

6.1 Deployment-Pipelines as Code

Phasen-, Gerichtete, Hierarische Pipelines, Steuerung von Pipelines

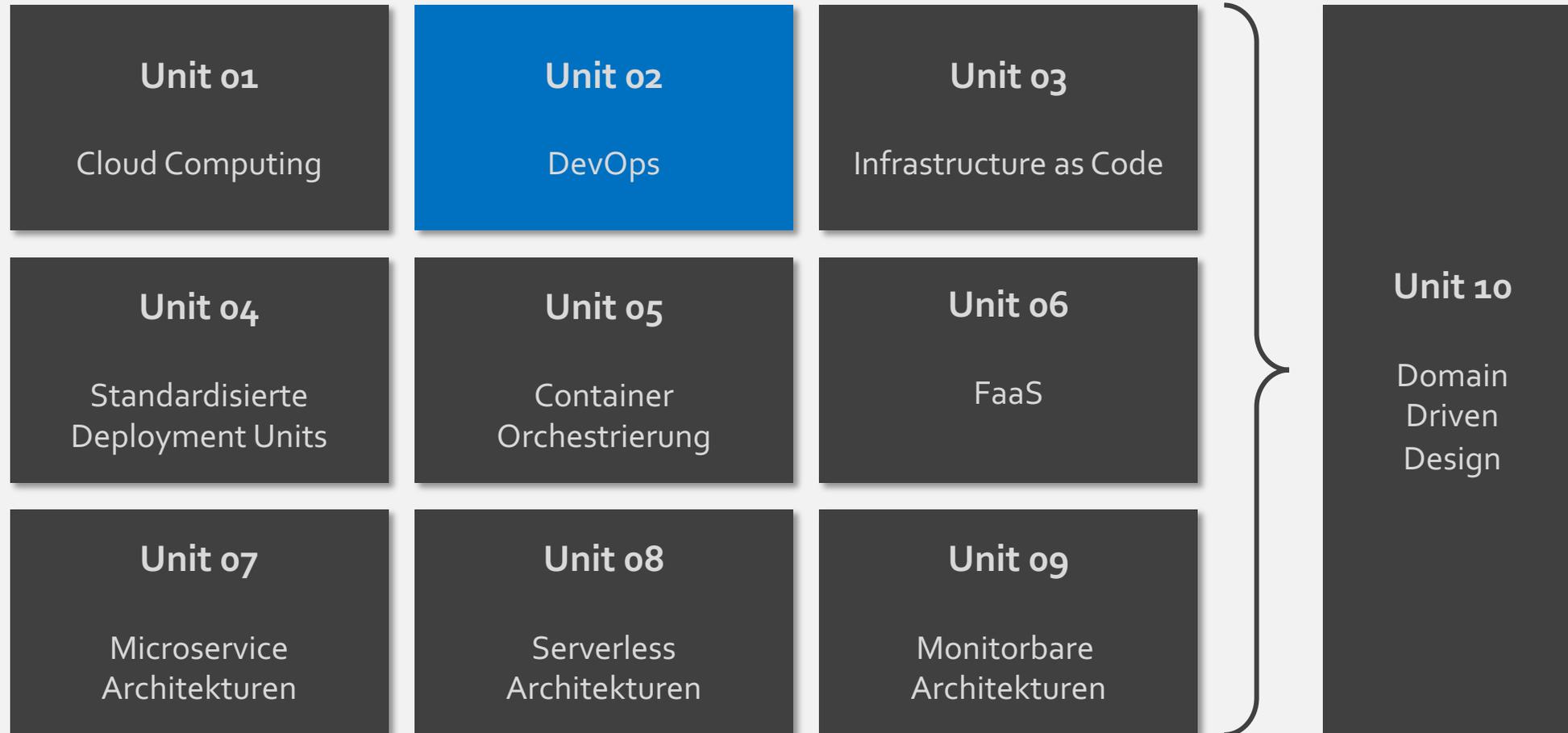
6.2 DevOps-geeignete Branching Strategien

Git-Flow, GitHub-Flow, Trunk-basierte Entwicklung

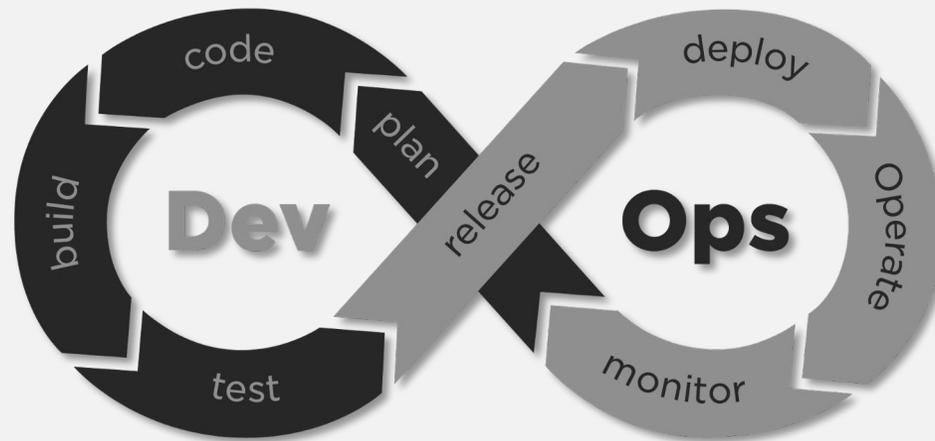
6.3 Zusammenfassung

INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls



- **DevOps Prinzipien**
- DevOps-Cycle konforme Architekturen und Umgebungen
- Continuous Begriffe und Deployment Pipelines



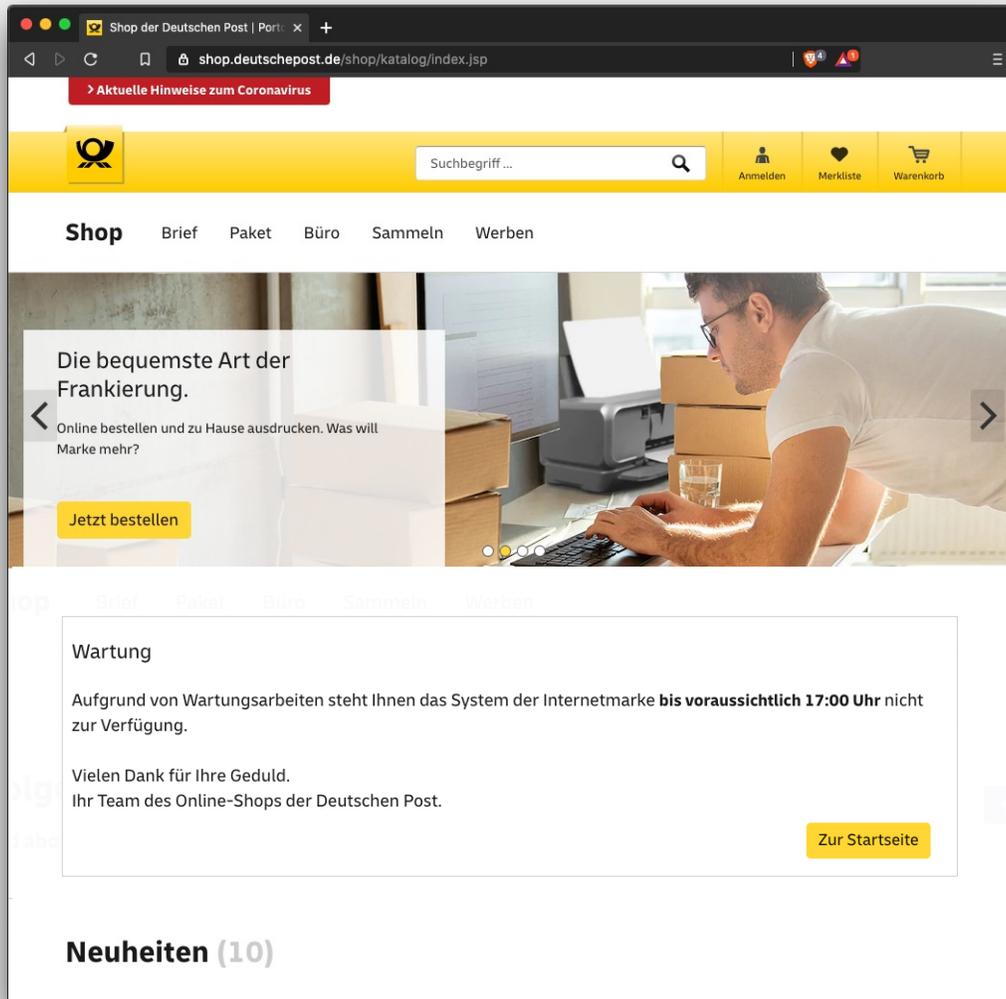
DAS PROBLEM



- Viele Banken datieren ihre IT in Rhythmen häufig einmal pro Quartal auf (und nennen das agil).
- Solche Update Zyklen können dann schon mal für alle Bankautomaten der Deutschen Bank mehrere Stunden, ggf. Tage dauern.
- Häufig werden solche Updates in Zeiten gelegt, in denen wenig Publikumsverkehr erwartet wird. Z.B. Sonntag morgens um 04:00 Uhr.
- Bei Bankautomaten akzeptieren wir das meist.
- Häufig gibt es auch einen Automaten einer anderen Bank (z.B. Sparkasse statt Deutsche Bank) in der Nähe, der zu einem anderen Zeitpunkt aktualisiert wird.

DAS PROBLEM

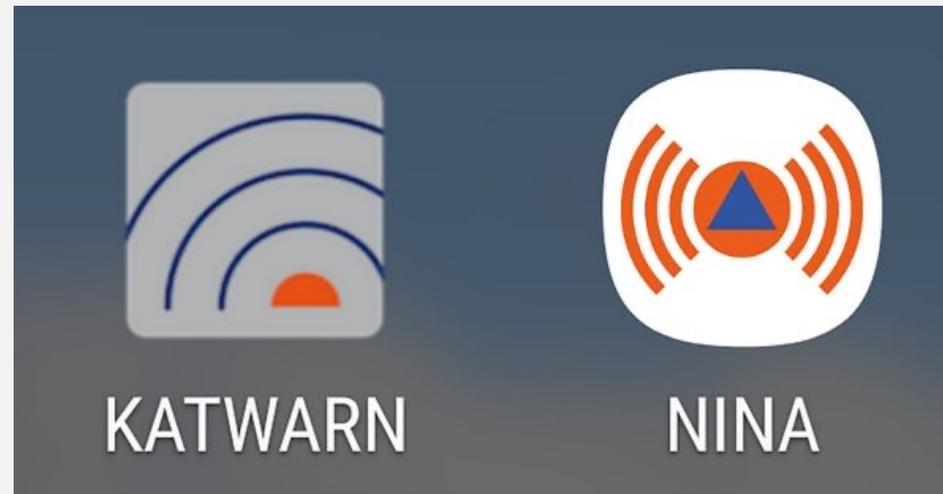
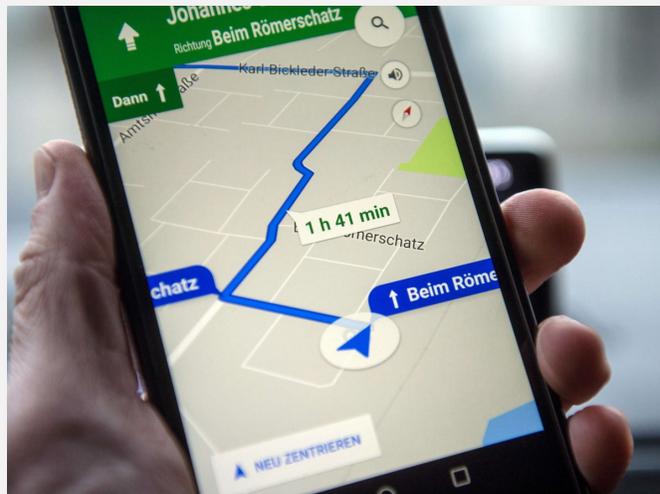
... hat die Deutsche Post übrigens bei E-Briefmarken offenbar auch.



Gesehen am: So., 20.09.2020, 11:45 Uhr

DAS PROBLEM

Aber würden Sie das bei diesen Diensten akzeptieren?

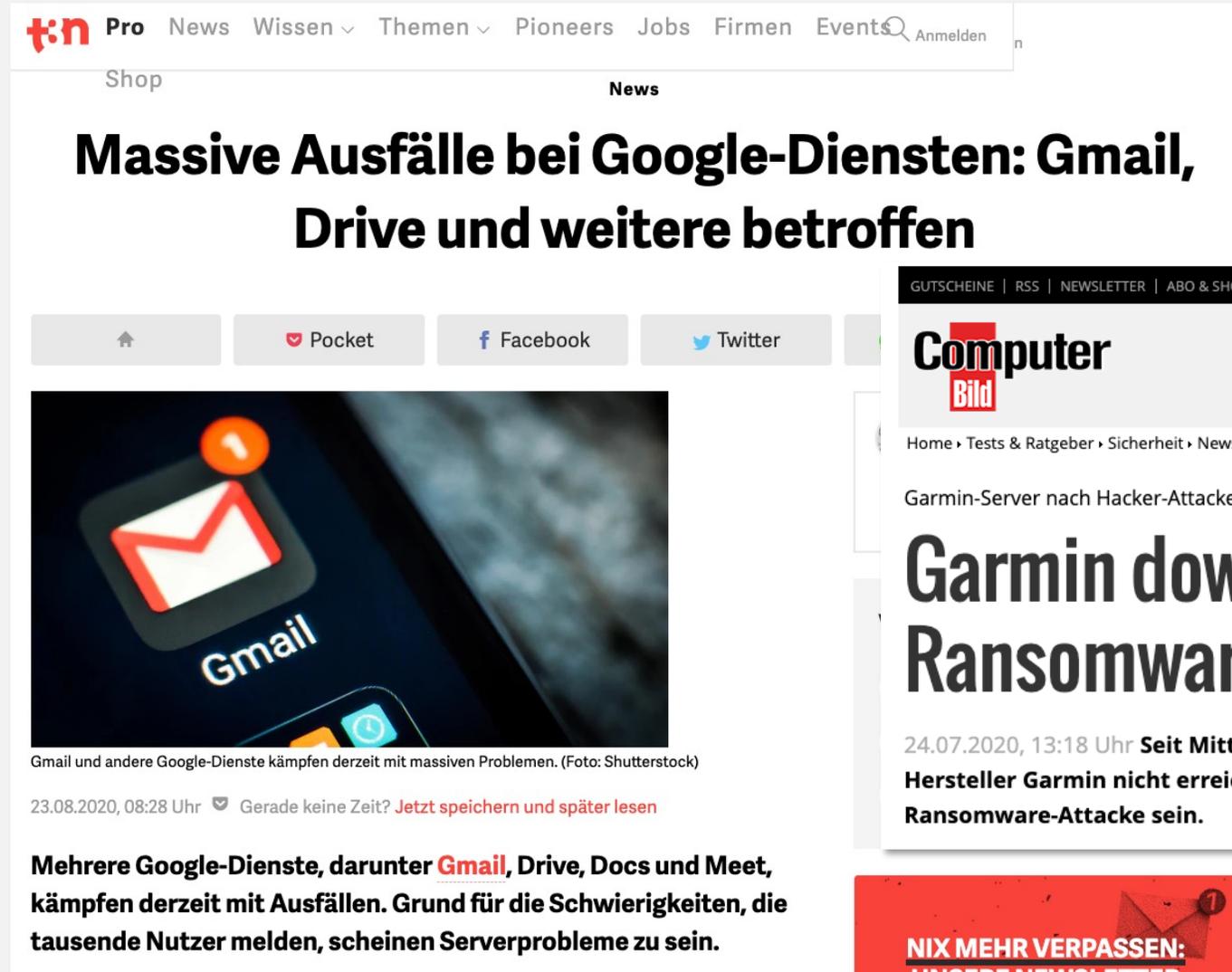


Verzeihung!

Unser Service steht Ihnen am kommenden Mittwoch von 07:00 Uhr bis 16:00 Uhr wegen eines weltweiten updates nicht zur Verfügung.

SCHLAGZEILEN

... die Minuten nach Problemaustritt in den Medien landen.



t3n Pro News Wissen Themen Pioneers Jobs Firmen Events Anmelden

Shop News

Massive Ausfälle bei Google-Diensten: Gmail, Drive und weitere betroffen



Gmail und andere Google-Dienste kämpfen derzeit mit massiven Problemen. (Foto: Shutterstock)

23.08.2020, 08:28 Uhr Gerade keine Zeit? Jetzt speichern und später lesen

Mehrere Google-Dienste, darunter **Gmail**, Drive, Docs und Meet, kämpfen derzeit mit Ausfällen. Grund für die Schwierigkeiten, die tausende Nutzer melden, scheinen Serverprobleme zu sein.

Solche Schlagzeilen kosten Geld, vor allem aber Renommee (und damit noch mehr Geld).



GUTSCHEINE | RSS | NEWSLETTER | ABO & SHOP | VIP-CLUB LOGIN

Computer Bild

START TESTS & RATGEBER BESTENLISTEN DOWNLOADS AKTIONEN

Suchbegriff eingeben SUCHEN

Home Tests & Ratgeber Sicherheit News

Garmin down: Knicken Server wegen Ransomware-Attacke ein?

24.07.2020, 13:18 Uhr Seit Mittwoch sind die Server von Smartwatch-Hersteller Garmin nicht erreichbar. Grund dafür könnte eine Ransomware-Attacke sein.

von Axel Palm

NIX MEHR VERPASSEN:
UNSERE NEWSLETTER

DAS DEVOPS PROBLEM



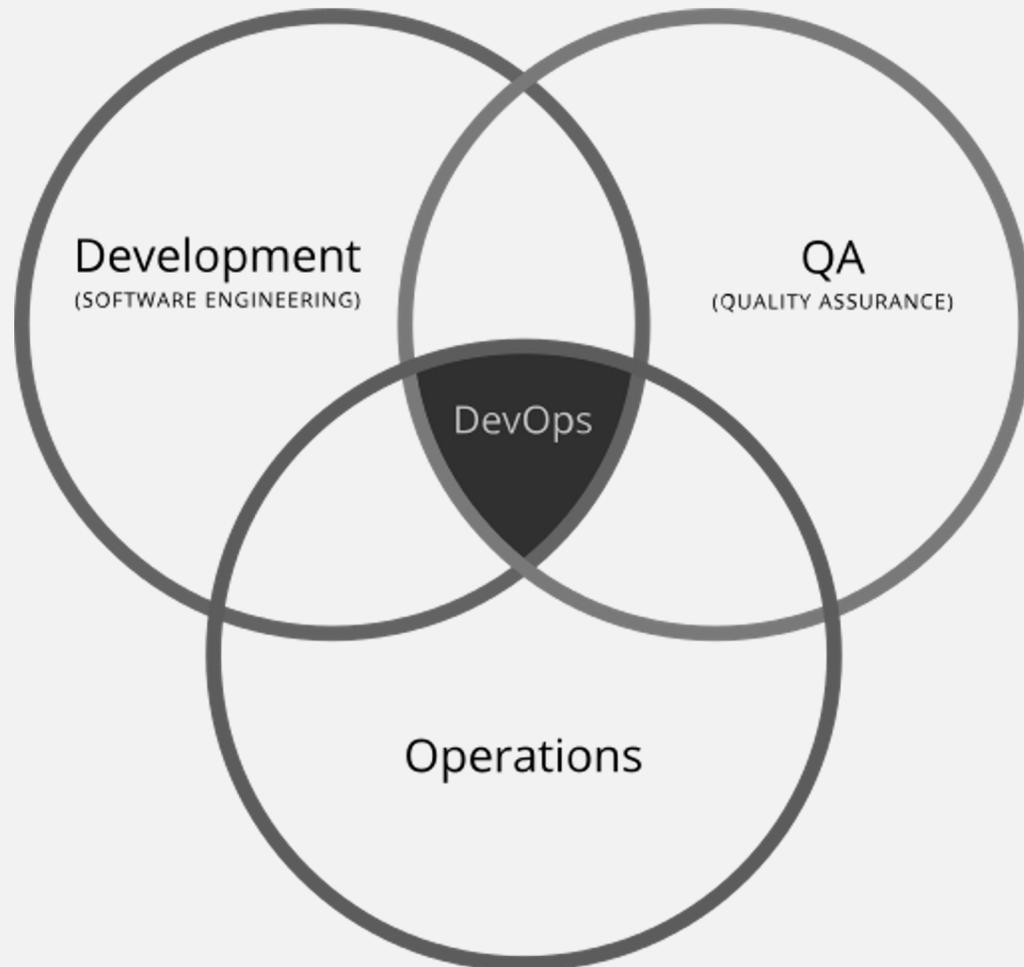
- Was machen Sie also, wenn Sie weltweit tätig sind (in allen Zeitzonen, in allen Kulturregionen)?
- Wenn Sie Ihre Dienstleistung ausschließlich online erbringen?
- Wenn Ihre Dienste überproportional häufig außerhalb der üblichen Geschäftszeiten nachgefragt werden (Video-Streaming, nachts, abends und am Wochenende)?
- Wenn Ihr Konkurrent nur einen Mausklick entfernt ist (Amazon Prime, Google Play, etc.)?
- Wenn Sie daher einfach keine Zeitfenster finden können, um weltweite Updates einzuspielen?

Angeblich spielt Amazon mehrere 100-mal pro Tag Updates in seine Produktivsysteme ein!

Als Vergleich: Bekannte Banken machen das etwa vier mal pro Jahr!

DEVOPS

Verbindet DEvelopment, Operations und Quality Assurance



DevOps ist die verbesserte Integration von **Entwicklung** und **Betrieb** durch mehr Kooperation und **Automation** mit dem Ziel Änderungen **schneller** in Produktion zu bringen.

DevOps ist vor allem eine Kultur (der Automatisierung und Verantwortung).

DEVOPS

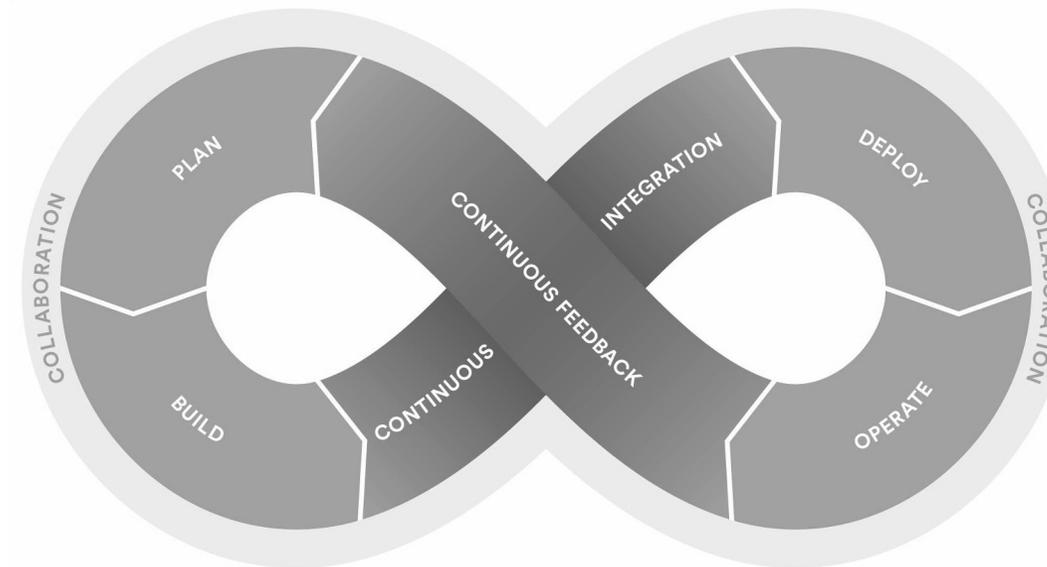
Der Versuch einer Definition

DevOps ist die Kultur, die Entwicklung (Dev) und den Betrieb (Ops) von Anwendungen besser miteinander zu integrieren, um

- kurze Releasezyklen,
- zuverlässige Inbetriebnahme und
- eine hohe Verfügbarkeit

zu erreichen.

Dies wird durch eine verbesserte Kooperation und einen höheren **Automatisierungsgrad** erreicht.



You build it, you run it!

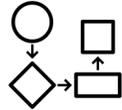
Damit DevOps allerdings auch reibungslos funktioniert, müssen Systeme konsequent darauf ausgelegt werden:

- Kultur
- Arbeitsorganisation
- Architektur
- Deployment Pipeline
- Deployment Environments
- Telemetriedaten

DEVOPS PRINZIPIEN

des Flow

Arbeit sichtbar machen



In der IT (anders als in der Produktion) kann Arbeit häufig per „Mausklick“ bewegt werden.

Weil das so einfach ist, werden manche Arbeitspakete zwischen Teams wie Ping Pong hin und her bewegt.

Um zu erkennen, wo Arbeit gut fließt und wo nicht, kann man auf visuelle Arbeitsboards (z.B. Kanban-Boards) zurückgreifen.

Dabei sollte immer die komplette Wertkette abgebildet werden, d.h. bis ein Feature tatsächlich produktiv läuft.

Work in Process beschränken



Unterbrechungen sind bei der Herstellung physischer Güter sehr kostenintensiv.

Studien haben gezeigt, dass sich die Zeit zum Erledigen selbst einfacher Aufgaben (wie dem Sortieren geometrischer Formen) beim Multitasking deutlich verlängert.

Multitasking sollte daher reduziert werden, um zu vermeiden, dass zwischen zu vielen Schritten und Kontexten hin und her gesprungen wird.

Flaschenhalse minimieren

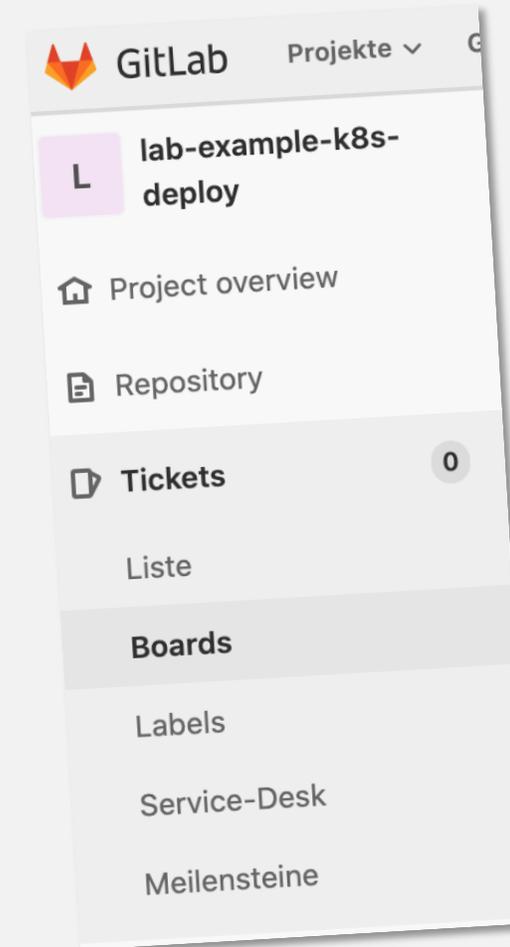
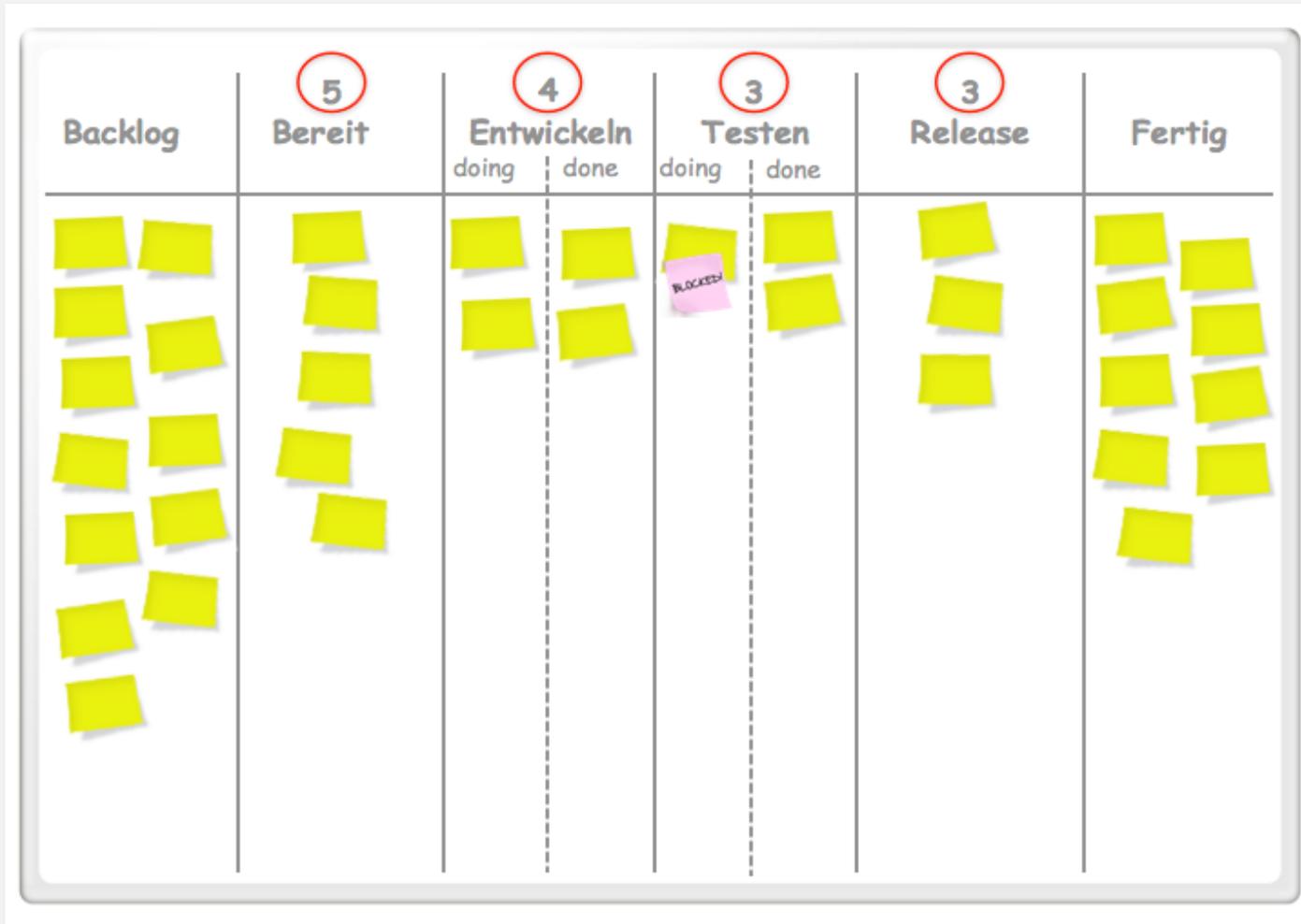


Flaschenhalse folgen häufig diesen Mustern:

- Langwieriges (manuelles) Erstellung von Test- und Produktiv-Umgebungen => Erstellung von Umgebungen im Self-Service auf Anforderung.
- Langwieriges (manuelles) Code-Deployment => Automatisiertes Deployment mittels Deployment Pipelines.
- Langwieriges (manuelles) Einrichten und Durchführen von Tests => Automatisiertes und parallelisiertes Testen, so dass die Testrate mit der Code-Entwicklungsrate mithält.
- Zu stark gekoppelte Architektur (lokale Änderungen müssen mit vielen abgestimmt werden) => Lose gekoppelte Architektur (z.B. Microservices) um Änderungen sicherer und mit mehr Autonomie durchführen zu können.

ARBEIT SICHTBAR MACHEN

Beispiel: Kanban-Board



Das ganze kann man natürlich auch elektronisch unterstützen.

In GitLab können Sie bspw. in jedem Projekt ihre Tickets als ein solches Board darstellen.

Probieren Sie es aus!

DEVOPS PRINZIPIEN

des Feedback

Probleme früh erkennen



Komplexe Systeme sind nicht vollständig von einer einzelnen Person überschau- und verstehbar.

Daher sollten Annahmen, die beim Design getroffen wurden, kontinuierlich geprüft werden. Z.B. mit Chaos Engineering, um Vertrauen in die Fähigkeit des Systems aufzubauen, turbulenten und unerwarteten Bedingungen standzuhalten. Ferner benötigt man für Feedback-Schleifen im Produktivsystem kontinuierlich und automatisiert erhobene Telemetriedaten.

Probleme sofort lösen



Problembehebungen im Produktivsystem sollten höhere Priorität als die weitere Entwicklung von Features bekommen.

Die Entwicklungspipeline sollte gestoppt werden.

Probleme sollten nie umgangen werden oder deren Lösung verschoben werden, um zu vermeiden, dass sich ein Problem fortsetzt und zu exponentiell steigendem Aufwand für dessen Behebung zu einem späteren Zeitpunkt führt.

Probleme professionell verantworten

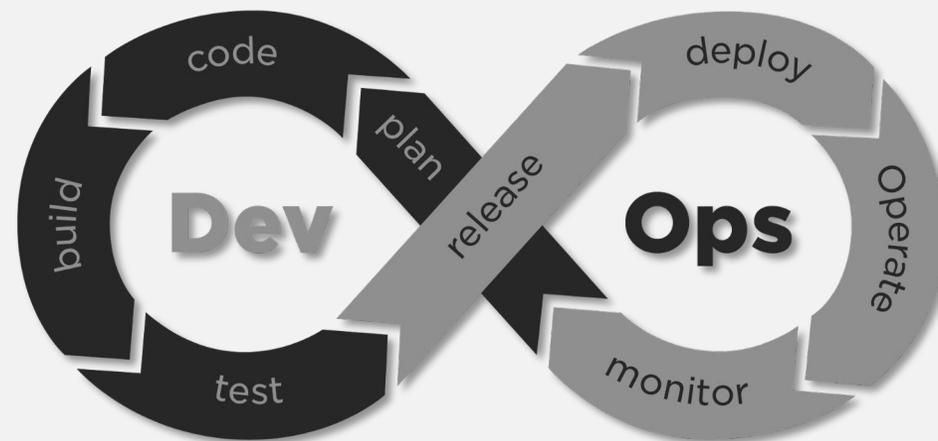


Erstaunlicherweise erhöht in komplexen Systemen das Hinzufügen zusätzlicher Kontrollschritte die Wahrscheinlichkeit für zukünftige Fehler. Entscheidungen müssen von Personen getroffen werden, die weit weg vom Problemraum sind.

Besser ist es, wenn Entwickler Verantwortung auch für den operativen Betrieb haben (you build it, you run it).

Dadurch liegt die Verantwortung für Qualität, Zuverlässigkeit und Entscheidungsbefugnis dort, wo auch die Arbeit erledigt wird und die meiste technische Expertise vorhanden ist.

- DevOps Prinzipien
- **DevOps-Cycle konforme Architekturen und Umgebungen**
- Continuous Begriffe und Deployment Pipelines



KONTAKT

Disclaimer

Nane Kratzke

📞 +49 451 300-5549

✉ nane.kratzke@th-luebeck.de

🌐 kratzke.mylab.th-luebeck.de

