



DevOps ILV - Aufgabenstellung 2

Hochschule Burgenland
Studiengang MCCE
Sommersemester 2025

Harald Beier* Susanne Peer[†] Patrick Prugger[‡]

Philipp Palatin[§]

5. Mai 2025

*2410781028@hochschule-burgenland.at

[†]2410781002@hochschule-burgenland.at

[‡]2410781029@hochschule-burgenland.at

[§]2310781027@hochschule-burgenland.at

Inhaltsverzeichnis

| | | | |
|----------|---|----------|----------|
| 1 | Aufgabenstellung | 2 | 3 |
| 1.1 | Release Management | | 3 |
| 1.2 | Beschreibung der Abkürzungen | | 3 |
| 1.3 | Vorteile von Infrastructure as Code (IaC) | | 4 |
| 1.4 | Unterschied DevOps Engineer vs. Site Reliability Engineer (SRE) | | 4 |
| | Literaturverzeichnis | | 6 |

1 Aufgabenstellung 2

1.1 Release Management

Aufgabe: *Beschreibe die Besonderheiten von Release Management im Sinne von SAFe und inwiefern es bei dem Unternehmen aus der Gruppenarbeit anwendbar wäre.*

Das Scaled Agile Framework (SAFe) bietet einen strukturierten Ansatz zur Skalierung agiler Methoden in größeren Organisationen. Im Zentrum steht der Agile Release Train (ART), ein langfristiges Team-of-Teams, das in regelmäßigen Program Increments (PI) von 8-12 Wochen gemeinsame Releases plant und koordiniert. Das *Release on Demand*-Prinzip ermöglicht flexible, aber zielorientierte Auslieferungen.

Die Continuous Delivery Pipeline bildet die technische Basis von SAFe mit drei Hauptphasen: *Continuous Exploration*, *Continuous Integration* und *Continuous Deployment*. Diese Pipeline visualisiert den durchgängigen Fluss von der Ideenfindung, bis zum Release. [Atlassian, 2024, APWide, 2023]

- Einführung regelmäßiger, teamübergreifender PI-Zyklen (8-12 Wochen) für gemeinsame Planung, Entwicklung und Auslieferung
- Bildung eines funktionsübergreifenden Teams mit geteilter Verantwortung für Releases
- Benennung eines Koordinators für Kommunikation und Dokumentation
- Implementierung einer einfachen Continuous Delivery Pipeline mit automatisierten Builds, Tests und Deployments

Diese gezielte Anwendung von SAFe-Prinzipien würde manuelle Prozesse reduzieren, konsistente Qualität sicherstellen und das *"Release on Demand"*-Prinzip ermöglichen. ABC Ad Tech würde effizienter und agiler arbeiten können, ohne das komplette Framework implementieren zu müssen, und gleichzeitig eine solide Grundlage für zukünftiges Wachstum schaffen.

1.2 Beschreibung der Abkürzungen

Aufgabe: *Was bedeuten die Abkürzungen MTTF, MTTR, RTO, MTTD, MT-TA und MTBF und wo liegen die Unterschiede? Betrachtung soll aus der DevOps Perspektive erfolgen.*

- **MTTF (Mean Time To Failure)** bezeichnet die durchschnittliche Zeit zwischen dem Start eines Systems und seinem Ausfall. In DevOps-Umgebungen misst MTTF, wie lange Anwendungen oder Services ohne Fehler laufen. Eine höhere MTTF deutet auf stabilere Software, als auch Infrastruktur hin. DevOps-Teams arbeiten daran, diese Zeit durch kontinuierliche Tests, Code-Reviews und Qualitätssicherung zu verlängern.

- **MTTR (Mean Time To Recovery/Repair)** ist die durchschnittliche Zeit, die benötigt wird, um ein System nach einem Ausfall wiederherzustellen. In DevOps ist MTTR besonders wichtig, da sie die Effizienz der Wiederherstellungsprozesse und Automatisierung widerspiegelt. Niedrigere MTTR-Werte zeigen eine verbesserte Resilienz. Durch Automatisierung, Überwachung und gut dokumentierte Runbooks können DevOps-Teams die MTTR reduzieren.
- **RTO (Recovery Time Objective)** ist das Ziel für die maximale Zeit, innerhalb derer ein System nach einem Ausfall wiederhergestellt werden soll. Im Gegensatz zu MTTR ist RTO ein vorab definiertes Ziel und keine Messung der tatsächlichen Leistung. DevOps-Teams definieren RTOs basierend auf Geschäftsanforderungen und implementieren dann Prozesse und Technologien, um diese Ziele zu erreichen.
- **MTTD (Mean Time To Detect)** misst die durchschnittliche Zeit zwischen dem Auftreten eines Fehlers und seiner Erkennung. In DevOps-Umgebungen ist eine schnelle Fehlererkennung entscheidend. Durch umfassende Überwachung, Logging und Alerting-Systeme streben Teams danach, die MTTD zu minimieren. Je früher ein Problem erkannt wird, desto schneller kann es behoben werden.
- **MTTA (Mean Time To Acknowledge)** bezieht sich auf die durchschnittliche Zeit zwischen der Erkennung eines Problems und dem Beginn der Behebungsmaßnahmen. Diese Metrik reflektiert die Reaktionsgeschwindigkeit des Teams. In DevOps-Kulturen mit klaren Verantwortlichkeiten und effektiven On-Call-Rotationen, wird die MTTA minimiert, was zu schnelleren Problemlösungen führt.
- **MTBF (Mean Time Between Failures)** ist die durchschnittliche Zeit zwischen zwei aufeinanderfolgenden Ausfällen eines reparierbaren Systems. MTBF umfasst sowohl die Betriebszeit (MTTF) als auch die Reparaturzeit (MTTR): $MTBF = MTTF + MTTR$. In DevOps-Umgebungen ist eine hohe MTBF wünschenswert, da sie auf stabile und zuverlässige Systeme hindeutet.

[Echolon, 2022, AlertOps, 2024]

1.3 Vorteile von Infrastructure as Code (IaC)

Aufgabe: *Inwiefern könnte der IaC Ansatz Vorteile bringen, um die zuvor genannten KPI's zu verbessern?*

1.4 Unterschied DevOps Engineer vs. Site Reliability Engineer (SRE)

Aufgabe: *Stelle den Unterschied zwischen einem DevOps Engineer und einem Site Reliability Engineer dar. Der Fokus liegt auf den Unterschieden hinsichtlich*

Aufgaben, benötigtem Know-how und dem Mehrwert für das Unternehmen.

Literaturverzeichnis

- [AlertOps, 2024] AlertOps (2024). Mtttd vs. mttf vs. mtbf vs. mttr comparison. Accessed 05.05.2025.
- [APWide, 2023] APWide (2023). Scaled agile framework: How safe release management works? Accessed 05.05.2025.
- [Atlassian, 2024] Atlassian (2024). Werte und prinzipien des scaled agile framework (safe). Zugriff am 05.05.2025.
- [Echolon, 2022] Echolon (2022). Was bedeuten die störungs- oder vorfallmetriken: Mtbfd, mttr. Zugriff am 05.05.2025.