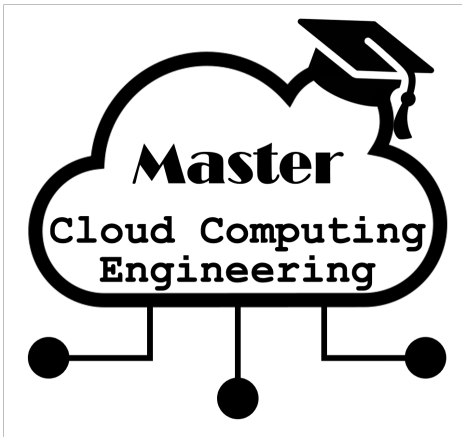

DevOps ILV - Aufgabenstellung 2

Hochschule Burgenland
Studiengang MCCE
Sommersemester 2025

Harald Beier* Susanne Peer† Patrick Prugger‡
Philipp Palatin§

13. Mai 2025

*2410781028@hochschule-burgenland.at
†2410781002@hochschule-burgenland.at
‡2410781029@hochschule-burgenland.at
§2310781027@hochschule-burgenland.at

Inhaltsverzeichnis

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]
Bei ABC Ad Tech existiert aktuell eine fragmentierte Struktur:

- Das Entwicklungsteam (AD-Dev) erstellt Softwareversionen auf einem einzelnen PC und kopiert Artefakte manuell auf eine Netzwerkfreigabe
- Das Betriebsteam (AD-Ops) führt alle zwei Wochen Deployments für Test- und alle drei Monate für Produktionsumgebungen durch
- Die Qualitätssicherung (Ad-QS) testet eigenständig und kommuniziert Feedback in Meetings

Diese Struktur weist erhebliche Mängel auf: *manuelle Übergaben, Zeitverluste, erhöhtes Fehlerpotenzial und unklare Verantwortlichkeiten.* Es fehlen gemeinsame Planungszyklen, kontinuierliche Integration und eine technische Grundlage für regelmäßige Releases. Obwohl SAFe primär für größere Unternehmen konzipiert ist, könnte ABC Ad Tech von ausgewählten Prinzipien profitieren:
[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, MTTA 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?*

Infrastructure as Code (IaC) bietet entscheidende Vorteile zur Optimierung der DevOps-Zuverlässigkeitsmetriken: *MTTF, MTTR, RTO, MTTD, MTTA und MTBF.*

Die **Mean Time To Failure (MTTF)** verbessert sich durch standardisierte, getestete Infrastrukturkonfigurationen. Da IaC die Implementierung automatisierter Tests ermöglicht, werden potenzielle Probleme frühzeitig erkannt. Die Versionskontrolle stellt sicher, dass bewährte Konfigurationen wiederverwendet werden können, was zu stabileren Systemen führt.

Die **Mean Time To Recovery (MTTR)** wird drastisch reduziert, da fehlerhafte Systeme schnell und zuverlässig, neu bereitgestellt werden können. Statt zeitaufwändiger manueller Fehlerbehebung ermöglicht IaC die automatisierte Wiederherstellung, mit einem einzigen Befehl. Die Idempotenz von IaC-Tools garantiert konsistente Wiederherstellungsprozesse.

Ambitionierte **Recovery Time Objectives (RTO)** werden durch automatisierte Infrastrukturbereitstellung erreichbar. Komplette Umgebungen können innerhalb von Minuten wiederhergestellt werden, was auch Disaster-Recovery-Szenarien unterstützt. Multi-Region- und Multi-Cloud-Strategien werden durch IaC praktisch umsetzbar.

Die **Mean Time To Detect (MTTD)** verkürzt sich durch konsistente Implementierung von Überwachungs- und Logging-Komponenten. Alarme und Schwellenwerte als Code führen zu präziserer Problemerkennung und reduzieren Fehlalarme.

Die **Mean Time To Acknowledge (MTTA)** verbessert sich durch bessere Dokumentation und Kontextinformationen. Bereitschaftsteams haben sofortigen Zugriff auf aktuelle Systemarchitekturinformationen, und automatisierte Runbooks können relevante Diagnosedaten bereitstellen.

Das **Mean Time Between Failures (MTBF)** steigt durch die Kombination aus weniger Ausfällen und schnellerer Wiederherstellung. Techniken wie Canary Deployments und Blue-Green-Deployments werden durch IaC praktikabel, was zu stabileren Systemen führt. IaC ermöglicht somit durch Automatisierung, Konsistenz, Testbarkeit und Reproduzierbarkeit beträchtliche Verbesserungen aller DevOps-Zuverlässigkeitsmetriken. Dies führt zu höherer Systemverfügbarkeit, besserer Benutzererfahrung und letztendlich zu einem höheren Geschäftswert für Organisationen, die diesen Ansatz implementieren.
[Centron, 2023, AWS, 2023]

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.*

DevOps Engineers fokussieren sich primär auf die Optimierung des Softwareentwicklungsprozesses. Sie bauen CI/CD-Pipelines, automatisieren Deployments und verbessern den Workflow, zwischen Entwicklung und Betrieb. Sie implementieren Tools für kontinuierliche Integration, Delivery und Deployment und arbeiten eng mit Entwicklungsteams zusammen, um Code schneller und zuverlässiger in Produktion zu bringen.

Site Reliability Engineers (SREs) konzentrieren sich hingegen auf die Zuverlässigkeit und Performance von Produktionssystemen. Sie definieren und überwachen SLAs, SLOs und SLIs, implementieren Observability-Lösungen und entwickeln Automatisierungen zur Verbesserung der Systemstabilität. SREs sind stärker in Incident Response, Kapazitätsplanung und die Analyse von Systemausfällen involviert.

DevOps Engineers steigern die Entwicklungsgeschwindigkeit und Agilität. Sie reduzieren die Zeit bis zur Markteinführung neuer Features, verbessern die Codequalität durch Automatisierung und fördern zudem die Zusammenarbeit zwischen Teams. Ihr Hauptbeitrag liegt in der Effizienzsteigerung des Entwicklungsprozesses und der Reduzierung von Silos.

SREs maximieren die Zuverlässigkeit und Verfügbarkeit von Produktionssystemen. Sie minimieren Ausfallzeiten, verbessern die Benutzererfahrung durch stabile Services und reduzieren die Kosten für Incident Management. Ihr Hauptbeitrag liegt in der Sicherstellung, dass Systeme skalierbar, resilient und performant bleiben, auch bei wachsender Komplexität und steigender Last. Während beide Rollen überlappende Fähigkeiten haben und das gemeinsame Ziel verfolgen, zuverlässige Software schneller zu liefern, liegt der Unterschied im Fokus: DevOps Engineers optimieren den Weg zur Produktion, während SREs die Qualität des Produktionsbetriebs sicherstellen. [NetApp, 2024, Brokee, 2024]

Literaturverzeichnis

[AlertOps, 2024] AlertOps (2024). Mttd 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.

[AWS, 2023] AWS (2023). Benefits of implementing infrastructure as code (iac). Accessed 05.05.2025.

[Brokee, 2024] Brokee (2024). The key differences between sre vs devops engineers. Accessed 05.05.2025.

[Centron, 2023] Centron (2023). Infrastructure as code (iac) – eine einführung. Zugriff am 05.05.2025.

[Echolon, 2022] Echolon (2022). Was bedeuten die störungs- oder vorfallmetriken: Mtbf, mttr. Zugriff am 05.05.2025.

[NetApp, 2024] NetApp (2024). Was ist site reliability engineering? sre im vergleich zu devops. Zugriff am 05.05.2025.