

**DOZIERENDER: MAX MUSTERMANN**

# **DATA ENGINEERING**

## Wer sind Sie?

- Person
- Arbeitgeber
- Funktion bzw. Aufgabe im Unternehmen
- Fun Fact
- Vorkenntnisse? Erwartungen?



---

**Datensystem-Grundlagen**

---

**1**

---

**Datenverarbeitung „at Scale“**

---

**2**

---

**Microservices**

---

**3**

---

**Governance und Sicherheit**

---

**4**

---

**Verbreitete Cloud-Plattformen und -Dienste**

---

**5**

---

**DataOps**

**6**

---

**LEKTION 1**

# **DATENSYSYSTEM-GRUNDLAGEN**



- **daten-intensive Systeme** erklären
- **Eigenschaften** gut entwickelter daten-intensiver Systeme erläutern
- **Systemzuverlässigkeit (Reliability)** eines Systems quantifizieren
- unterschiedliche Ansätze zur **System-Skalierung (Scalability)** gegeneinander abwägen
- verschiedene Aspekte von **Wartbarkeit (Maintainability)** diskutieren und in der Entwicklung daten-intensiver Systeme berücksichtigen



1. Nennen Sie **drei Aspekte**, die es modernen datenintensiven Datensystemen ermöglichen, **große Datenmengen zu verarbeiten** und dabei **stabil** zu laufen.
2. Erläutern Sie **Möglichkeiten**, diese Systeme an **höheren oder niedrigeren Leistungsbedarf** anzupassen.
3. Erklären Sie, wie diese Systeme so gestaltet werden können, dass sie in Bezug auf Wartung und Weiterentwicklung **auf einfache Weise betrieben** werden können.

## Reliability

Systemzuverlässigkeit

- Reliability-Metriken
- Hardware-Fehler
- Software-Fehler
- menschliche Fehler

## Scalability

Skalierbarkeit

- Load (Systemlast) und Throughput (Durchsatz)
- Service Level Agreements (SLAs)
- Scale Up – Scale Out (Hoch- oder Ausskalieren)

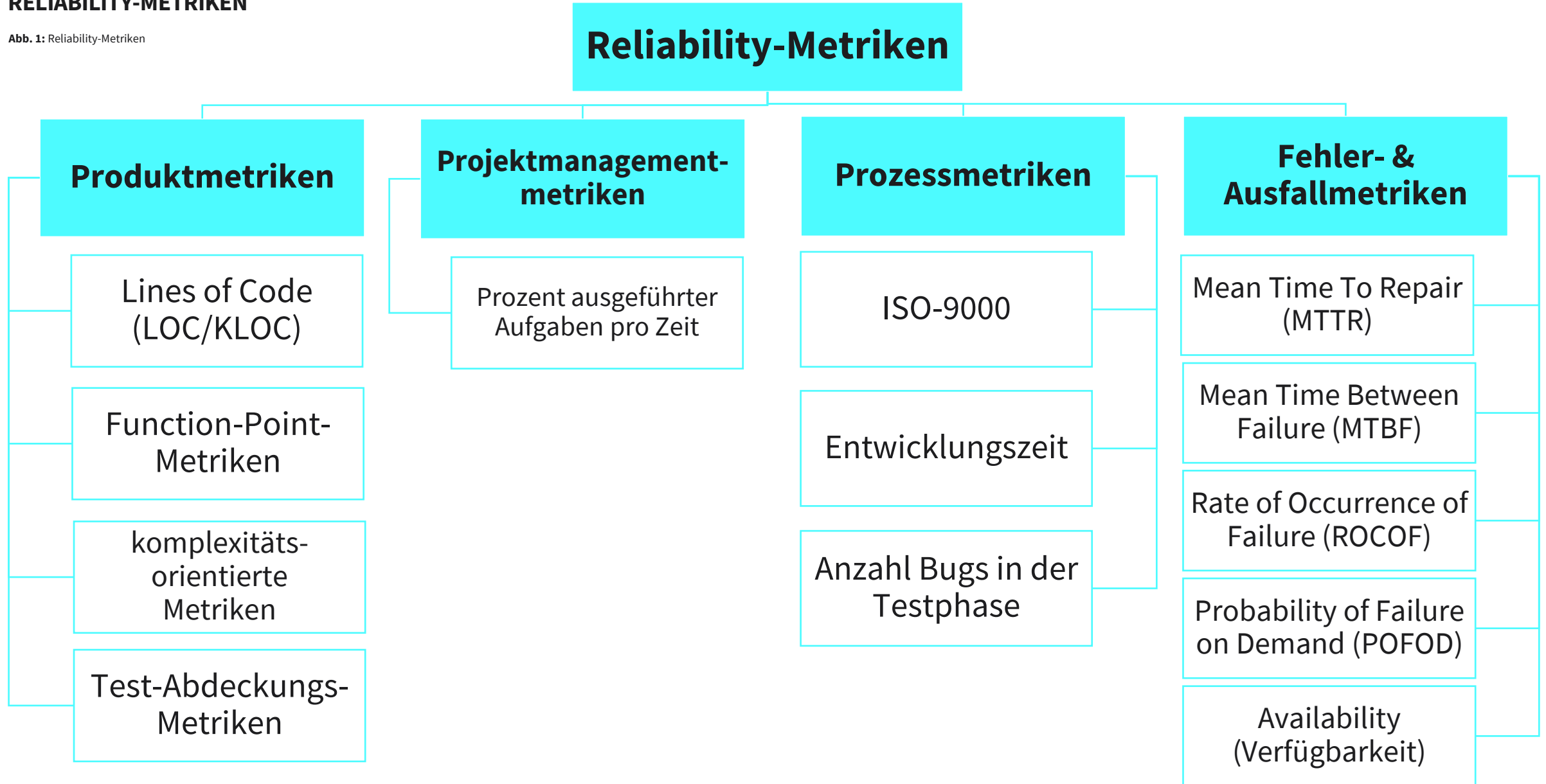
## Maintainability

Wartbarkeit

- Operabilität
- Einfachheit
- Evolvierbarkeit

## RELIABILITY-METRIKEN

Abb. 1: Reliability-Metriken

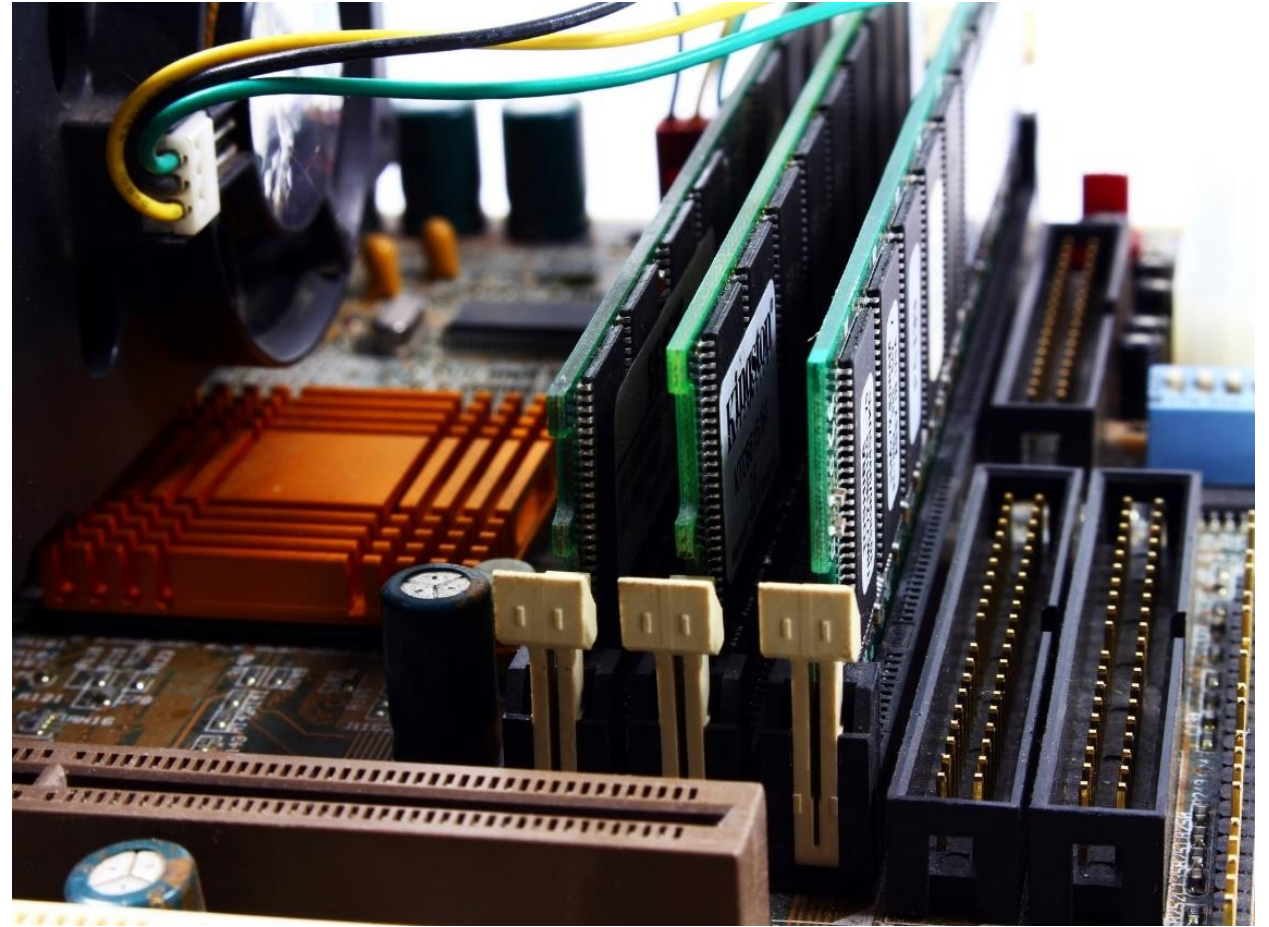




## HARDWARE-FEHLER

- physische Beschädigung einer Komponente innerhalb einer Maschine
- Redundanz (bspw. RAIDs)
- Redundanz über mehrere Maschinen hinweg
- Stromausfall / Netzwerk-Störungen
- Redundanz über mehrere Datenzentren/Verfügbarkeitszonen

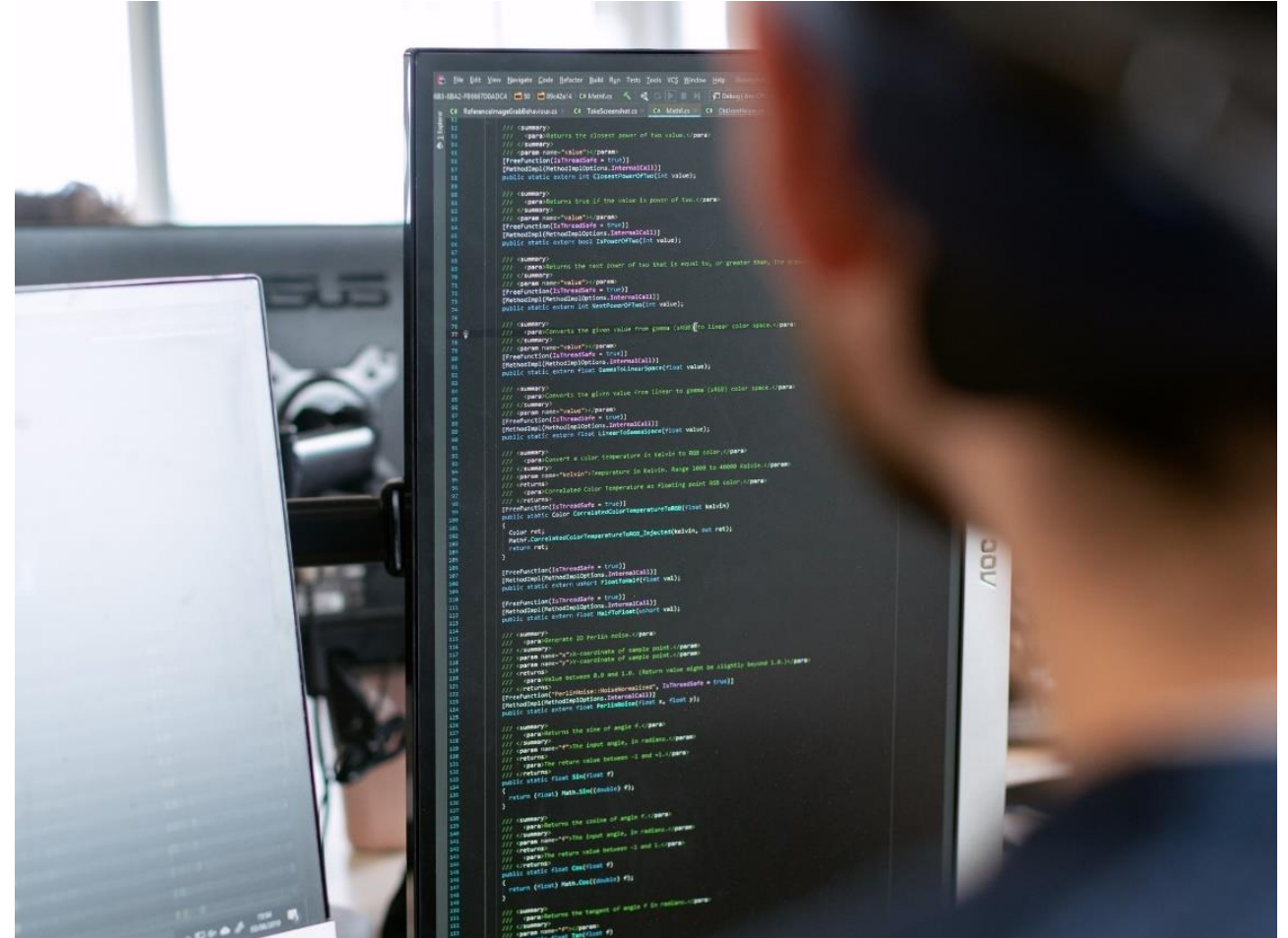
Abb. 2: Hardware



## SOFTWARE-FEHLER

- Memory Leakage
- Ausfall von Diensten
- Cascading Failure  
("Wasserfall-Fehler")
- Software ist ein **mentales Produkt** (Détienne, 2002),  
in das Menschen Fehler  
einbauen

Abb. 4: Software





## MENSCHLICHER FEHLER

- **Menschen** stellen ein erhebliches Potenzial dar, **Fehler** in ein System einzubringen
- **32%** aller sicherheitsrelevanter Vorfälle werden durch die **eigenen Mitarbeiter:innen** verursacht (BakerHostetler, 2017)
- effiziente **Recovery-Tools**
- **minimale Möglichkeit** bieten menschliche Fehler zu begehen
- intensives **Testen**

Abb. 6: Menschlicher Fehler



- **System Load** (Systemlast)

- Read-to-Write (Lese-Schreibe)-Verhältnis auf die Systemspeicherung
- Anzahl der Online-Nutzer:innen
- Anfrage pro Zeit

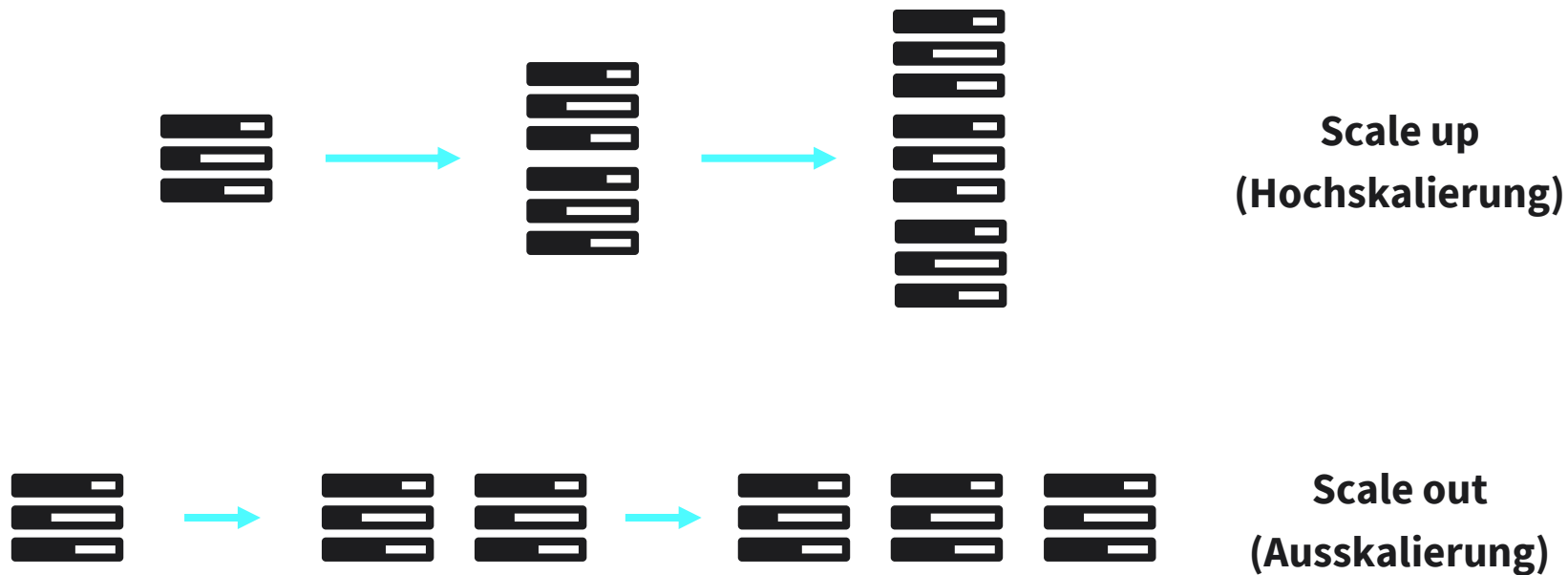
- **System Performance Load** (Leistungslast)

- Laufzeit für Lese-/Schreibe-Operationen
- Throughput (Durchsatz)
- I/O-Rate: Input-Output Operations per Second (IOPS)
- Response Time

## SCALE UP – SCALE OUT (HOCH- ODER AUSSKALIEREN)

- **Software-seitige** Optimierung
- **physische Ressourcen** skalieren

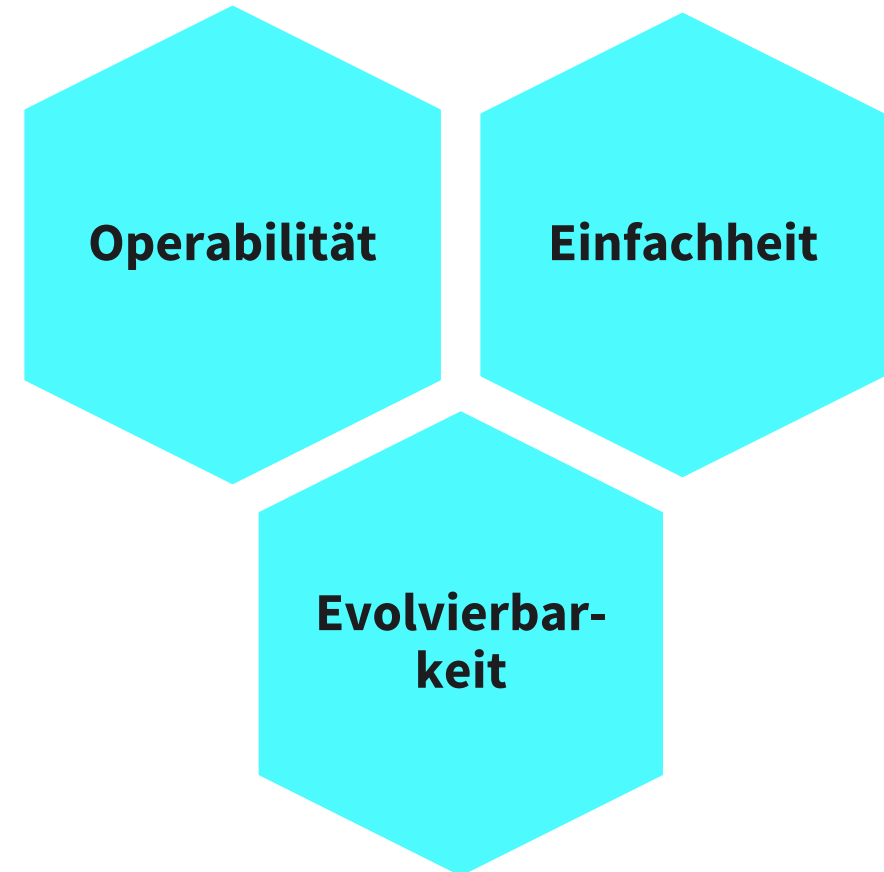
Abb. 7: Scale up – scale out



## MAINTAINABILITY (WARTBARKEIT)

- Wir machen es einfach, dass System zu **installieren, benutzen** und **modifizieren**
- Wir **minimieren** die Unannehmlichkeiten von **legacy maintenance** (Wartung alter Systeme)
  - Korrigierende Wartung
  - Adaptive Wartung
  - Perfektionierenden Wartung
  - Präventive Wartung

Abb. 8: Maintainability (Wartbarkeit)



**System-Operator:innen** sind verantwortlich dafür:

- die „Gesundheit“ des Systems zu **überwachen**
- **Sicherheits-Patches** zu installieren
- **potenzielle Probleme** vorherzusehen und zu lösen
- eine **stabile Produktionsumgebung** sicherzustellen
- **Wissen** über das System zu erhalten

Abb. 8: Maintainability (Wartbarkeit)



**Wir machen es** System-Operator:innen **einfacher**, indem wir ...

- transparente **Überwachungsmöglichkeiten** bereitstellen
- **Automatisierung und Integration** mit Standard-Werkzeugen unterstützen
- eine stichhalte **Dokumentation** anbieten
- sicherstellen, dass **Dienste/Maschinen unabhängig voneinander** funktionieren
- **Standardeinstellungen** und **Selbstheilungseigenschaften** im System implementieren

Abb. 8: Maintainability (Wartbarkeit)

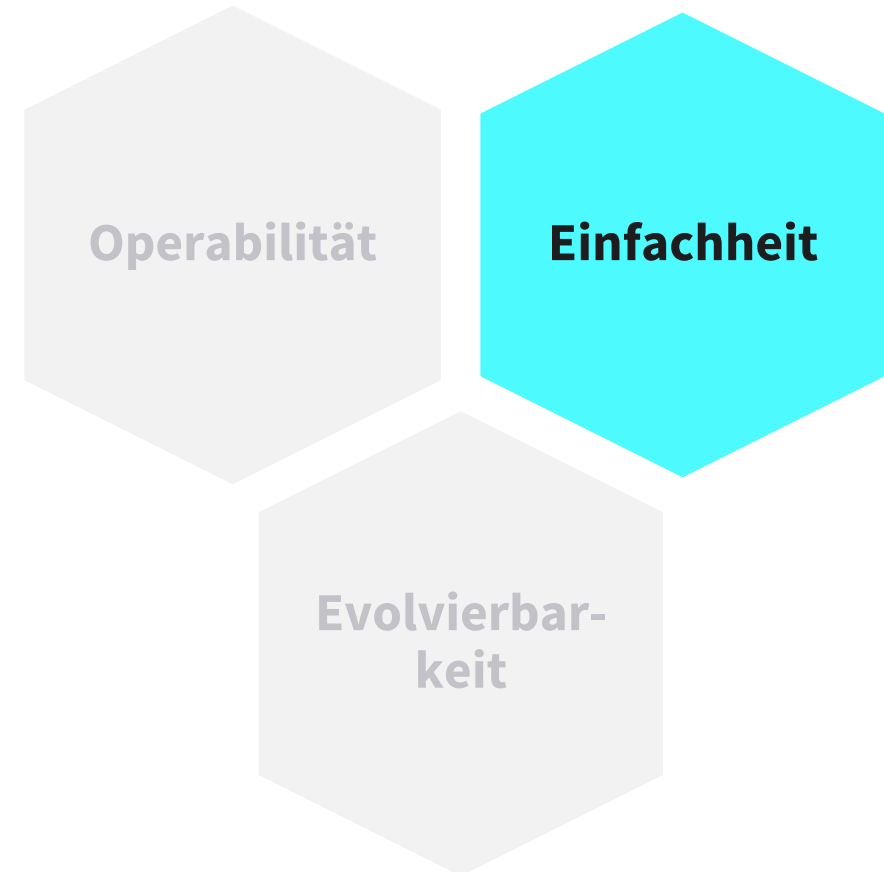




## EINFACHHEIT

- Wir gestalten den **Code so einfach wie möglich** und nur so komplex wie unbedingt nötig.
- Wir **reduzieren**:
  - ungewollte **Komplexität**
  - **langen** Code
  - **Modul-Interaktionen** und **Abhängigkeiten**
  - inkonsistente **Namensgebung**
- Wir erhöhen das **Level** an **Abstraktion**.

Abb. 8: Maintainability (Wartbarkeit)



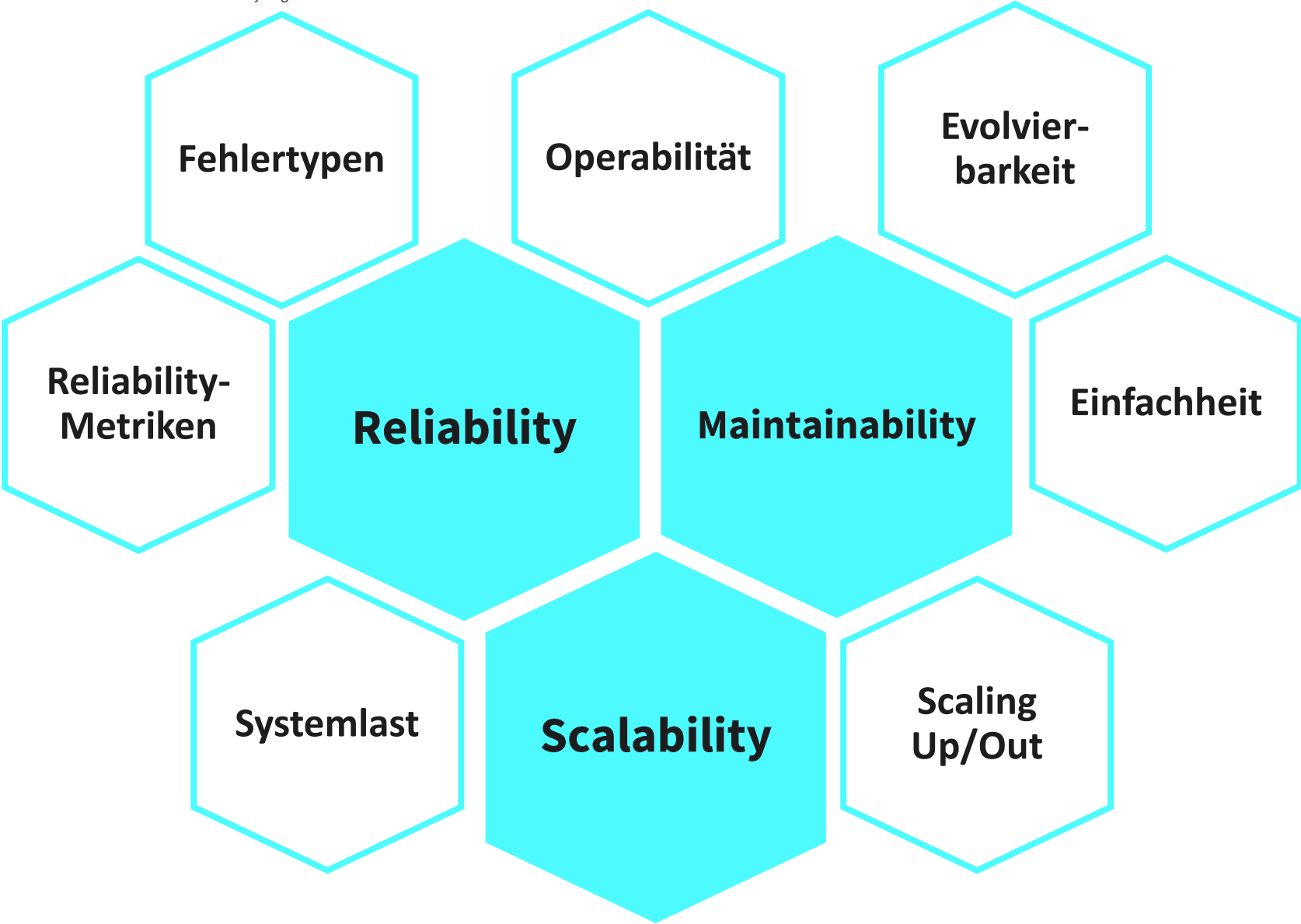
## EVOLVIERBARKEIT

- Wir machen es **so einfach wie möglich**, das System zu **verändern** und **weiterzuentwickeln**.
- ähnlich zum Begriff „**agil**“
- **DataOps**

Abb. 8: Maintainability (Wartbarkeit)



Abb. 9: Synergien





- **daten-intensive Systeme** erklären
- **Eigenschaften** gut entwickelter daten-intensiver Systeme erläutern
- **Systemzuverlässigkeit (Reliability)** eines Systems quantifizieren
- unterschiedliche Ansätze zur **System-Skalierung (Scalability)** gegeneinander abwägen
- verschiedene Aspekte von **Wartbarkeit (Maintainability)** diskutieren und in der Entwicklung daten-intensiver Systeme berücksichtigen

**EINHEIT 1**

# **TRANSFERAUFGABE**

## TRANSFERAUFGABE

Ein Start-Up, das **nachhaltige Produkte** in **kleineren Geschäften** vertreibt, war in den letzten Jahren sehr erfolgreich. Infolge sollen **weltweit weitere Filialen** eröffnet werden. Als Data Engineer:in sind Sie damit beauftragt, das **Datensystem zu entwerfen**, welches Daten über die **angebotenen Produkte und deren Zulieferer** speichert und verarbeitet.

Erarbeiten Sie im Team **Kernpunkte**, die sicherstellen sollen, dass dieses System **angemessen effektiv** und **performant** laufen wird. Überlegen Sie sich zu jedem dieser Punkte **konkrete Maßnahmen**, die im System implementiert werden sollen.

Bitte stellen Sie Ihre  
Ergebnisse vor.  
Im Plenum werden  
die Ergebnisse  
diskutiert.





1. Welche Aufgabe fällt nicht in die Zuständigkeit von Dateningenieur:innen?

- a) Analyse der Daten
- b) Entfernen beschädigter Daten
- c) Erfassung von Daten aus verschiedenen Quellen
- d) Optimieren von Datenbanken für die Analyse





2. Welche der folgenden Antwortmöglichkeit ist die Bezeichnung für die Wahrscheinlichkeit eines fehlerfreien Softwarebetriebs über einen bestimmten Zeitraum in einer bestimmten Umgebung?
- a) Zuverlässigkeit
  - b) Instandhaltbarkeit
  - c) Skalierbarkeit
  - d) Sicherheit



3. Wenn einige Fehler in der Anwendung gefunden wurden und diese behoben werden müssen, welche Art von Wartung wird dann eingesetzt?
- a) adaptive Wartung
  - b) perfektionierende Wartung
  - c) präventive Wartung
  - d) korrektive Wartung

## QUELLENVERZEICHNIS

- BakerHostetler. (2017, 9. Mai). *Webinar: Be Compromise Ready: Go Back to the Basics* [Video]. BakerHostetler. <https://www.bakerlaw.com/events/webinar-be-compromise-ready-go-back-to-the-basics>
- Détienne, F. (2002). *Software Design: Cognitive Aspects* (F. Bott, Hrsg.). Springer. <https://doi.org/10.1007/978-1-4471-0111-6>
- Hamilton, J. (2007, November). On Designing and Deploying Internet-Scale Services. In USENIX Association (Hrsg.), *Proceedings of the 21st conference on Large Installation System Administration Conference (LISA'07)* (S. 231–242).
- Hartz, M. A., Walker, E. L. & Mahar, D. (1997). *Introduction to software reliability: A state of the art review*. The Center.
- Naylor, W. & Joyner, B. (2014, Januar). A discourse on software safety and software reliability. 2014 Reliability and Maintainability Symposium, 1–5. <https://doi.org/10.1109/rams.2014.6798493>
- stevepb (2015). *Fehler Verschütten Ausrutscher Unfall Error* [Fotografie]. Pixabay. Abgerufen am 15. März 2022, von <https://pixabay.com/de/photos/fehler-versch%C3%BCtten-ausrutscher-876597/>
- ThisIsEngineering. (2020, 04. März). *Man in Black Shirt Sitting in Front of Computer* [Fotografie]. Pexels. <https://www.pexels.com/photo/man-in-black-shirt-sitting-in-front-of-computer-3861959/>
- Valentine Tanasovich. (2019, 30. Juni). *Black and Gray Computer Motherboard* [Fotografie]. Pexels. <https://www.pexels.com/photo/black-and-gray-computer-motherboard-2588757>
- Wilkins, D. J. (2002). The Bathtub Curve and Product Failure Behavior (Part 1 of 2). *Reliability HotWire*, 21. Abgerufen am 15. März 2022, von <https://www.weibull.com/hotwire/issue21/hottopics21.htm>

© 2022 IU Internationale Hochschule GmbH

Diese Inhalte sind urheberrechtlich geschützt. Alle Rechte vorbehalten.

Diese Inhalte dürfen in jeglicher Form ohne vorherige schriftliche Genehmigung der IU Internationale Hochschule GmbH nicht reproduziert und/oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.