Requirements- und Systems-Engineering für künstlich intelligente Systeme

Jan Ebert, HOOD GmbH

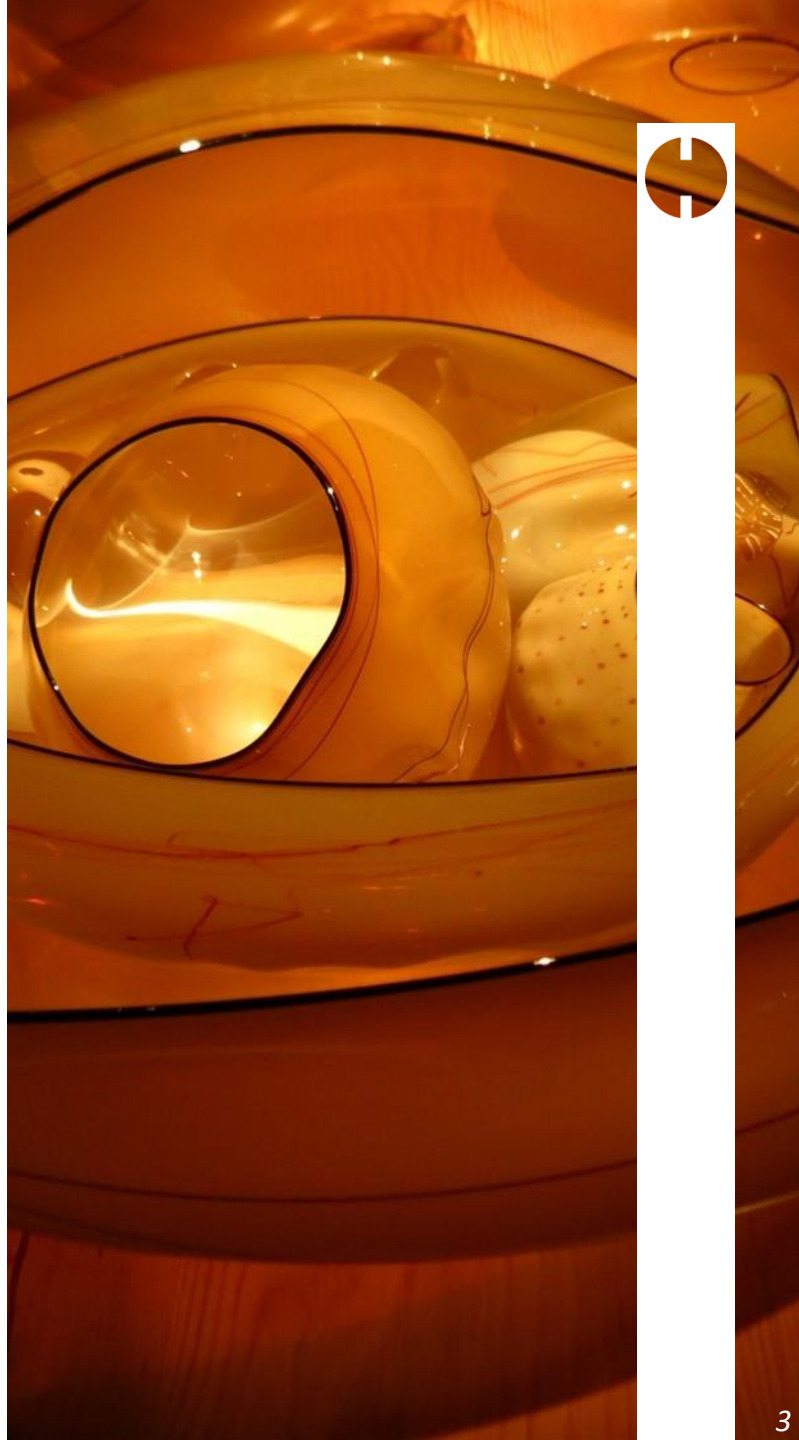
Scope

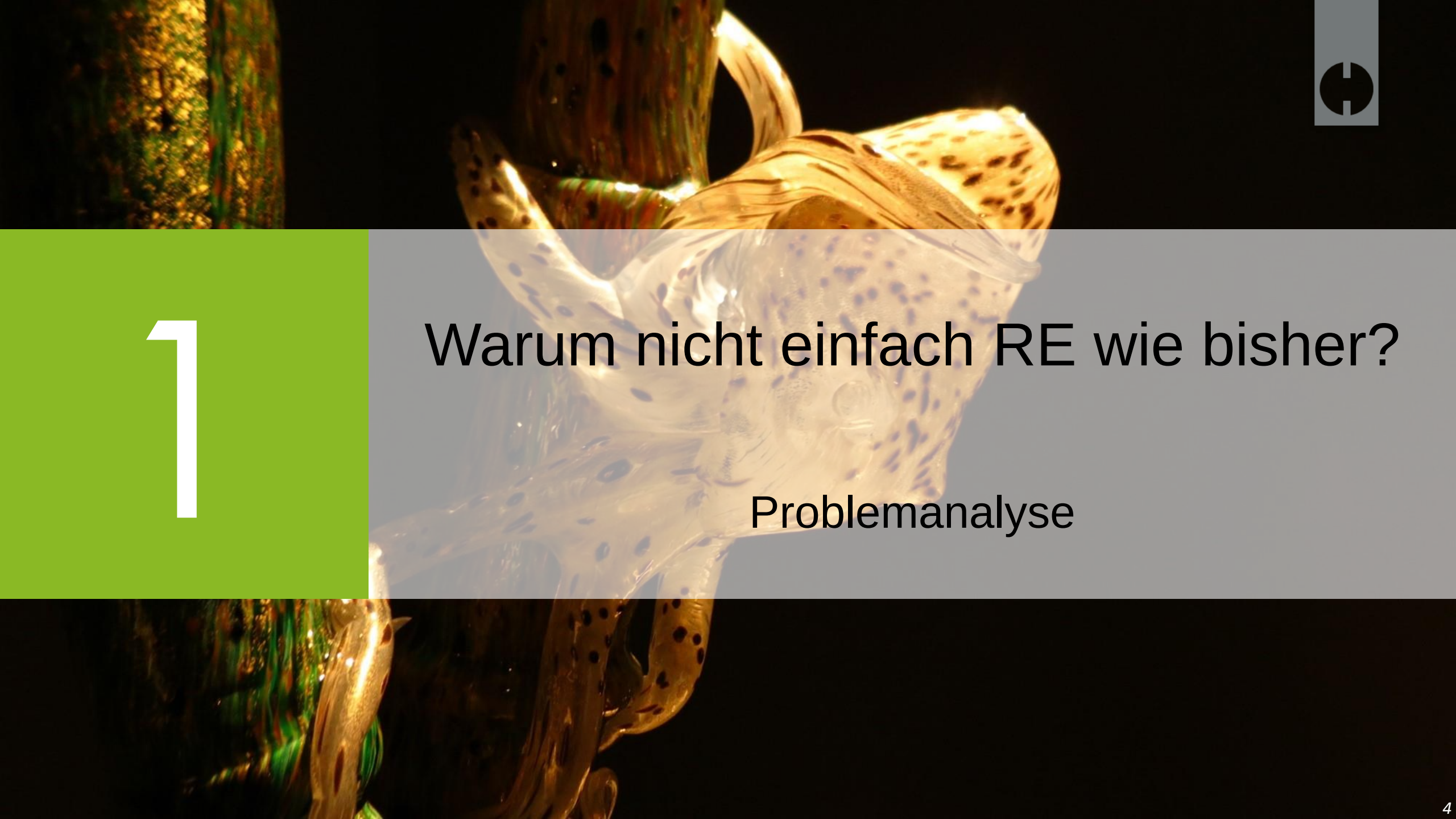
Es geht nicht darum, wie KI dabei helfen kann, das Requirements Engineering zu unterstützen.

Sondern darum Systeme, die KI/ML enthalten, zu spezifizieren.

Was sind „künstlich intelligente“ Systeme?

Definition KI:	„Automatisierung geistiger Arbeit“
(AI):	*Automation of mental tasks*
Ziel starker KI:	Die intellektuellen Fertigkeiten von Menschen
(strong/general AI)	zu erlangen oder zu übertreffen
Schwache KI:	Automatisierung konkreter Anwendungsprobleme
(weak/narrow AI)	mit den Methoden maschinellen Lernens (ML)
KI/ML Systeme:	Systeme, deren Verhalten durch ML basierte Komponenten unterstützt wird



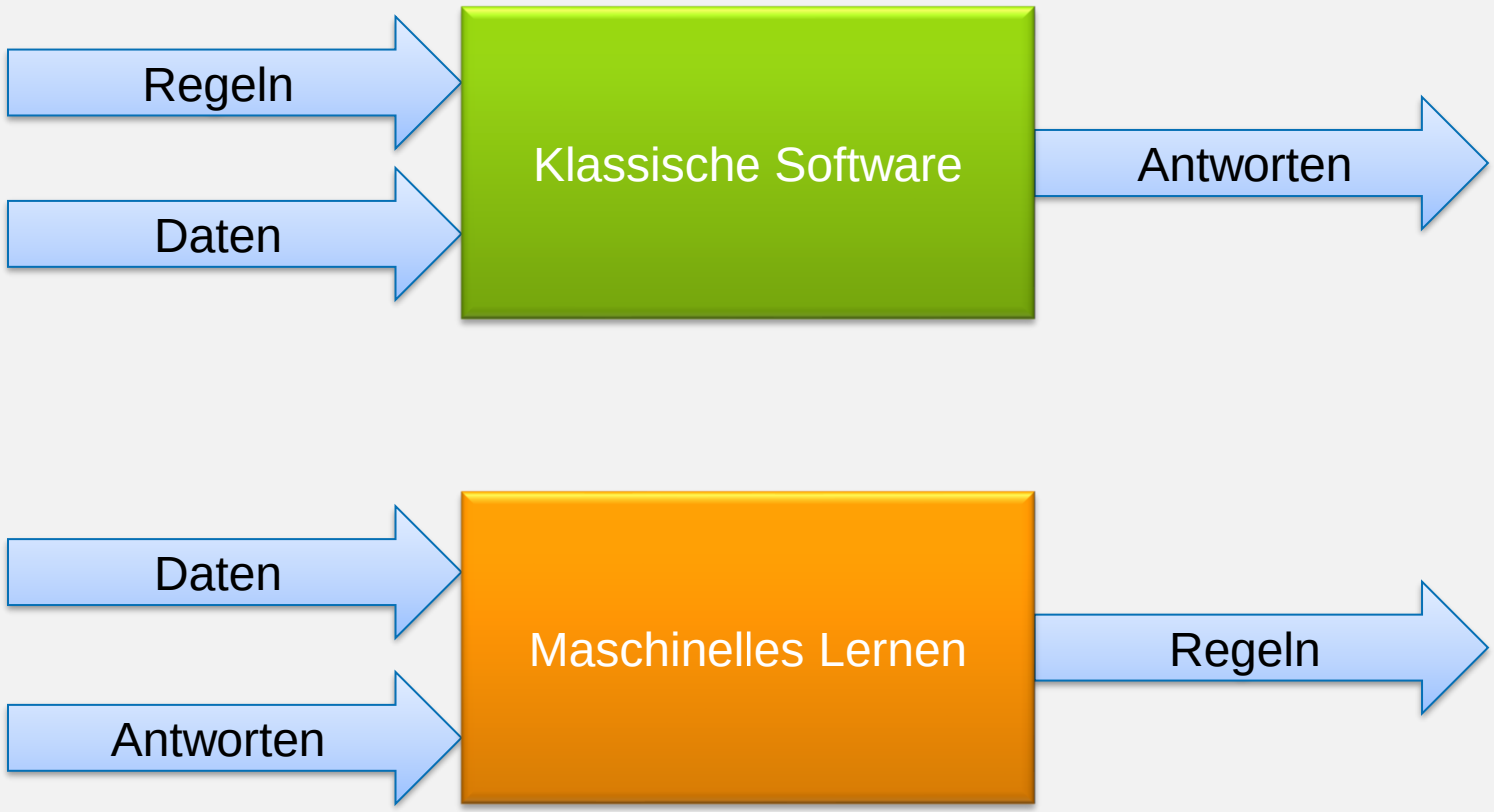


1

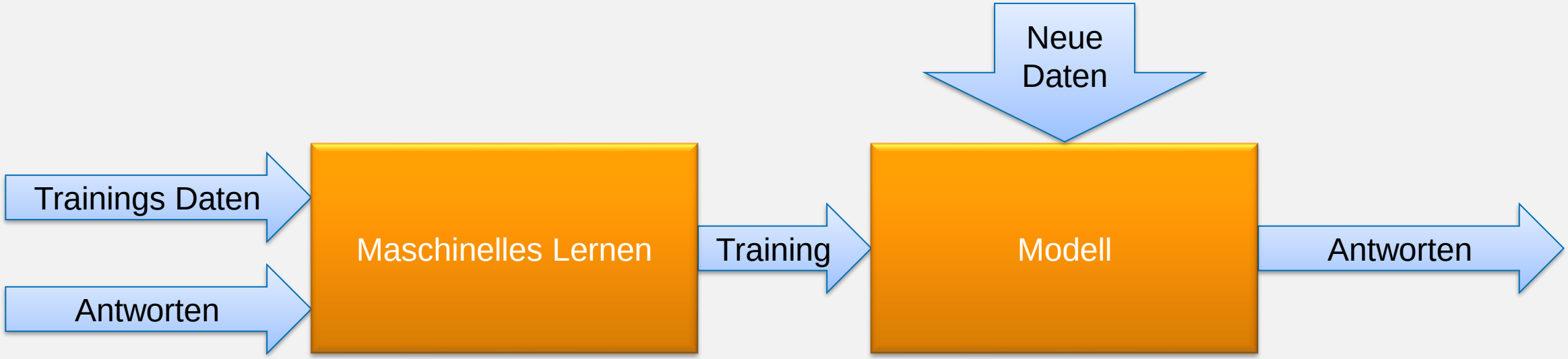
Warum nicht einfach RE wie bisher?

Problemanalyse

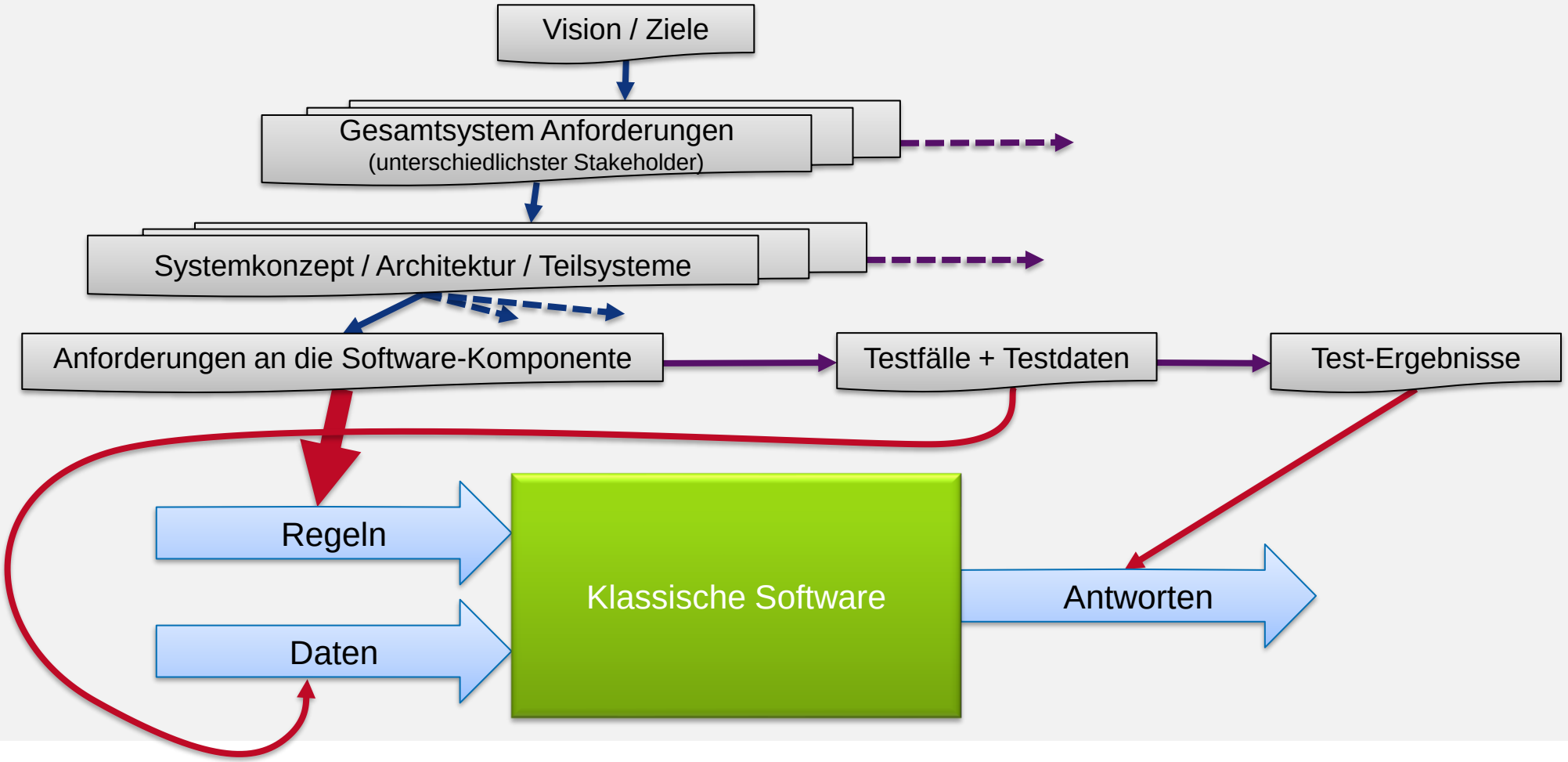
Der fundamentale Unterschied



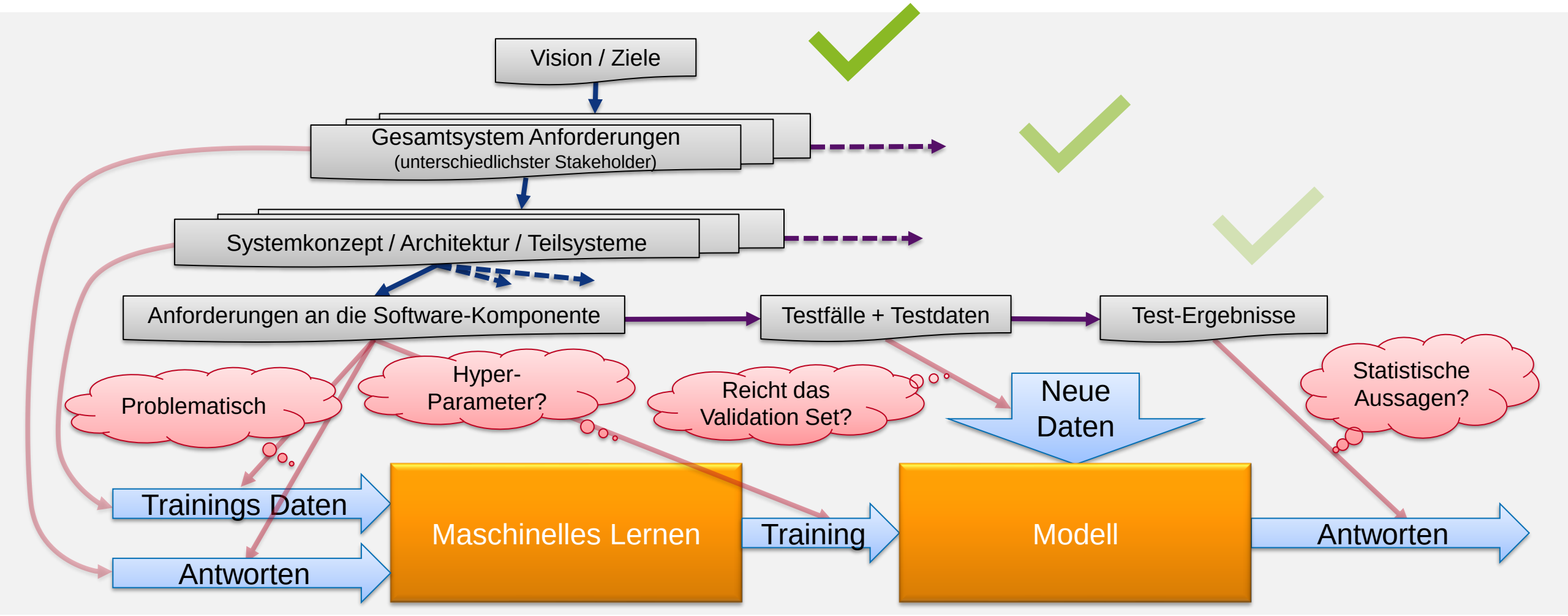
Genaugenommen…



Spezifikation und Verifikation klassischer Systeme



Lässt sich das auf KI/ML Systeme übertragen?

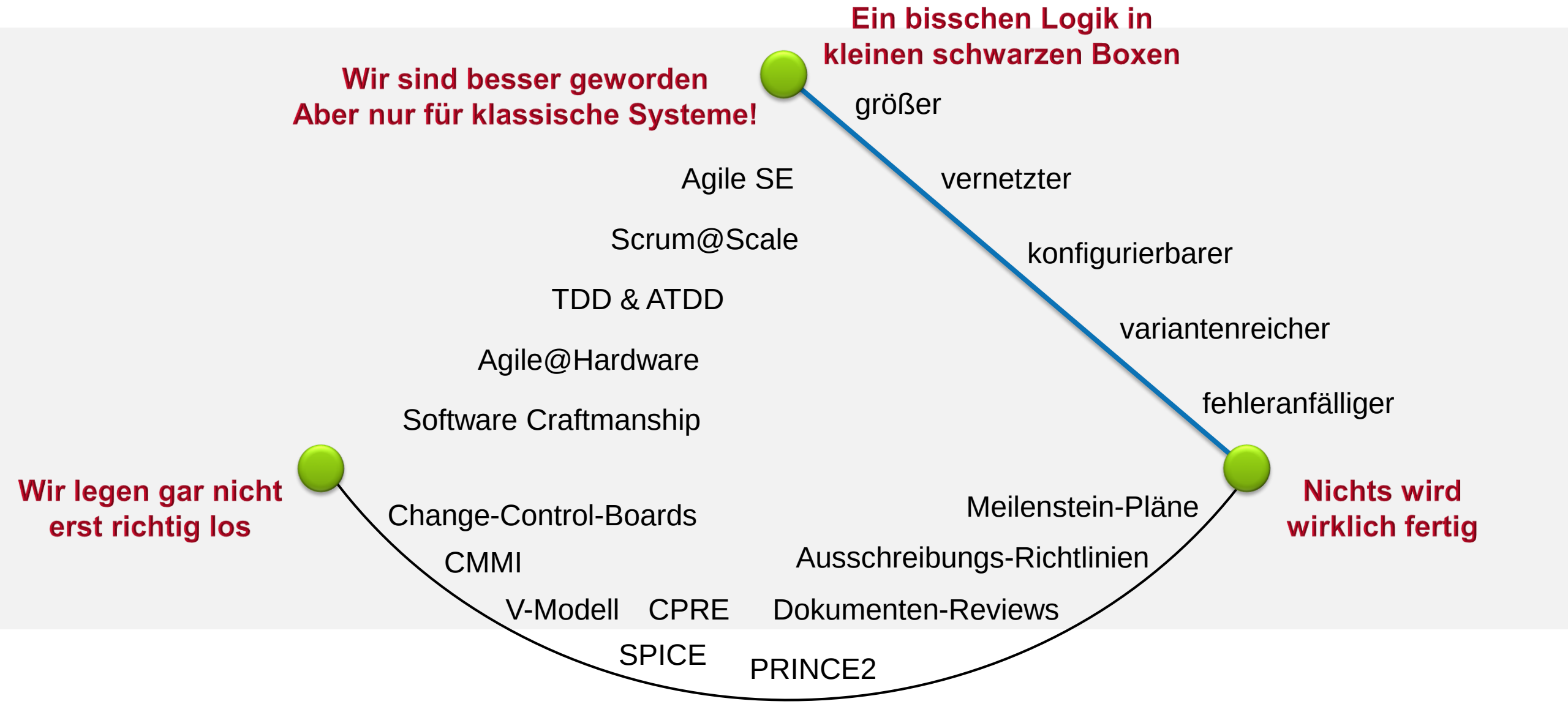


2

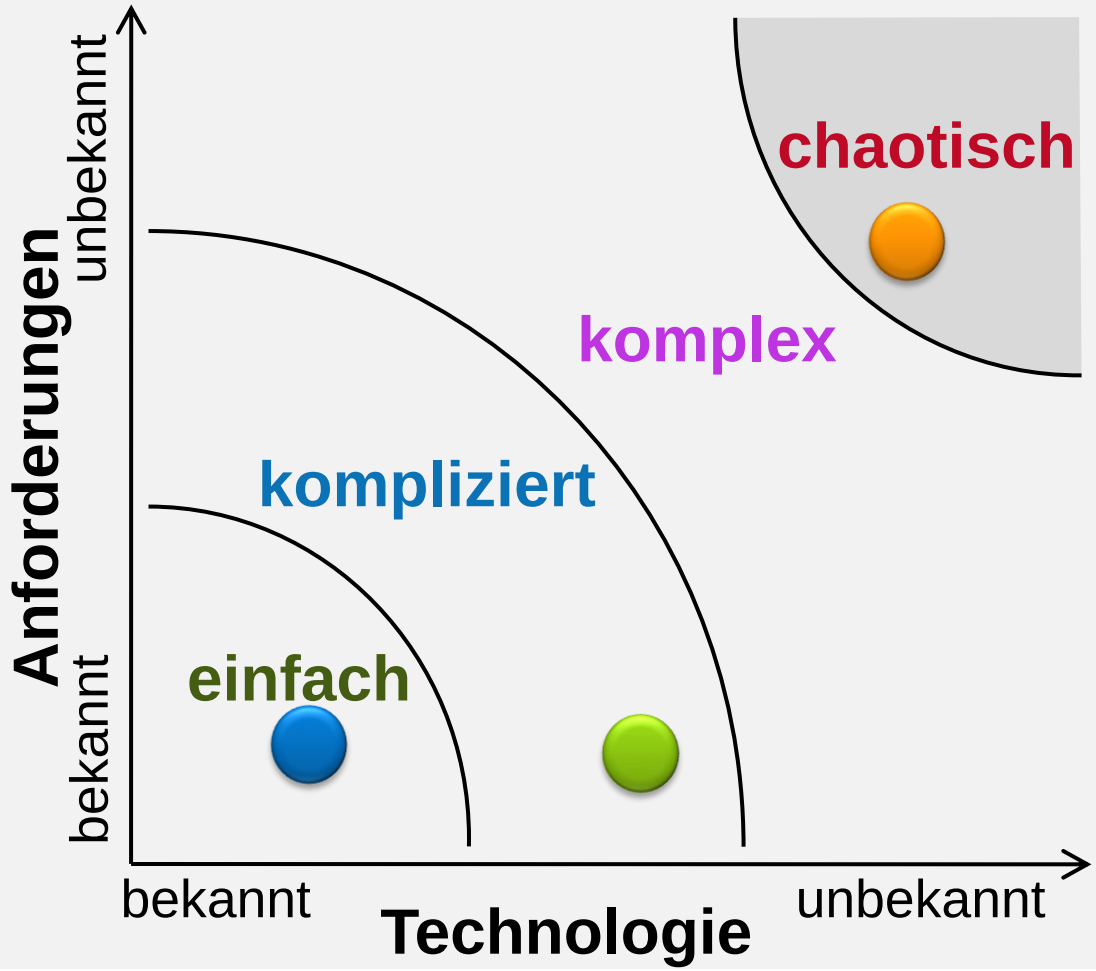
Wo stehen wir?

State of the Art

Historischer Rückblick: Als die Software einzog



Kulturkonflikte im Stacey Diagramm



- Maschinenbau / Ingenieurwesen ohne SW
- Klassisches Software + Systems Engineering
- Künstliche Intelligenz / Machine Learning

Die jeweils geeigneten Vorgehensweisen Passen nicht zusammen.

Wie tief ist der Graben?

Conferences > 2019 IEEE 27th International ...

Requirements Engineering for Machine Learning: Perspectives from Data Scientists

Publisher: IEEE | Cite This | PDF

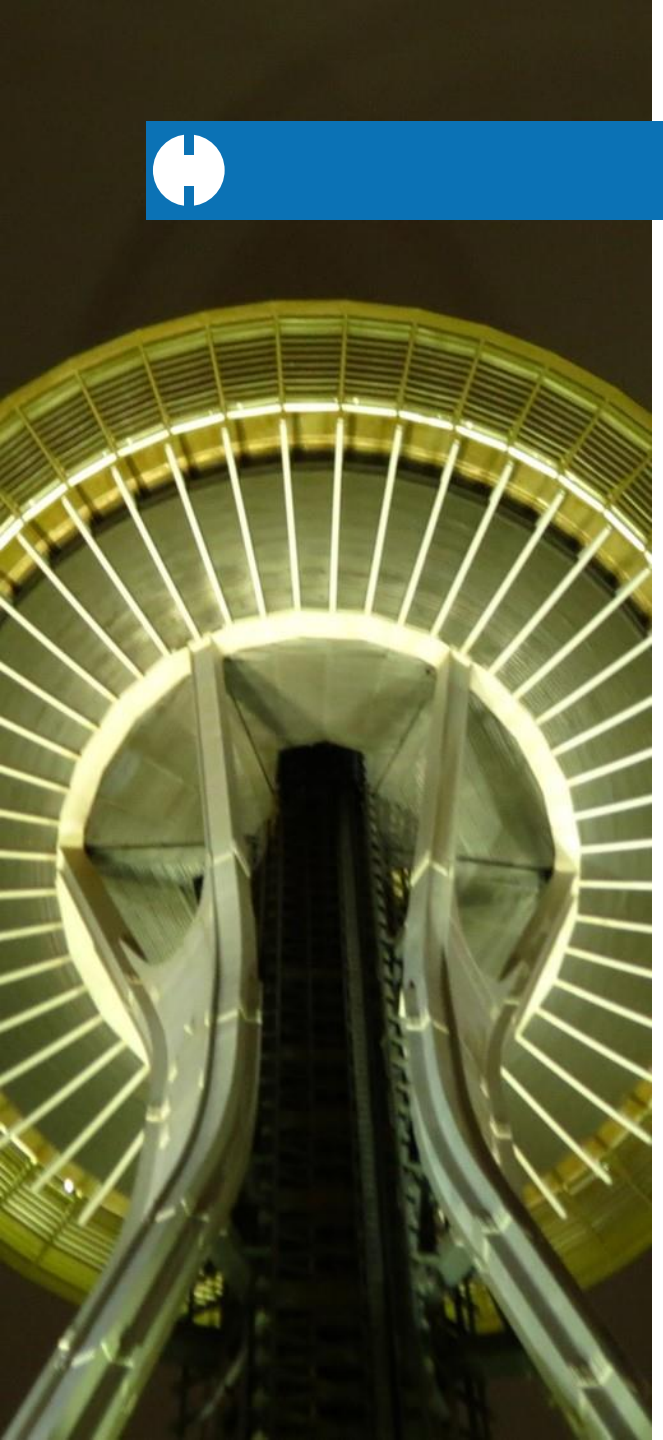
Andreas Vogelsang ; Markus Borg **All Authors**

Sophia Yang in Towards Data Science · Jan 5 ★

Don't Make Data Scientists Do Scrum

Thoughts from a data scientist and Certified Scrum Master — Scrum Guide: "Scrum is a lightweight framework that helps people, teams and organizations…

Data Science 8 min read



Zwischenfazit

- KI-Systeme werden zwar bereits erfolgreich, aber chaotisch entwickelt
- ML-Entwicklung ist noch nicht zu einer Engineering-Disziplin gereift
- Die Integration von ML in die Entwicklung von Systems-of-Systems gestaltet sich daher schwierig
- Die etablierten Methoden des (agilen) Software- und Systems-Engineering passen nicht für explorativ arbeitende ML Startups
- Spezifikation und Verifikation von KI-Systemen kann prinzipiell nicht genau so funktionieren wie für regelbasierte Systeme
- **Wir brauchen neue Praktiken, um KI-Systeme effizient und zuverlässig spezifizieren und implementieren zu können**

3

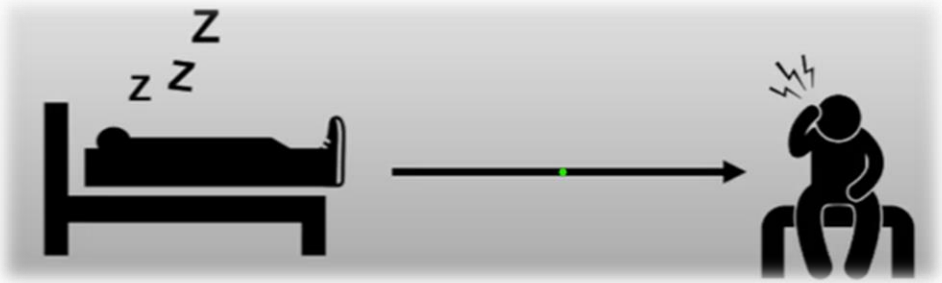
Maßnahmen

Lösungsansätze und Empfehlungen

Maschinelles Lernen ist kein Selbstzweck

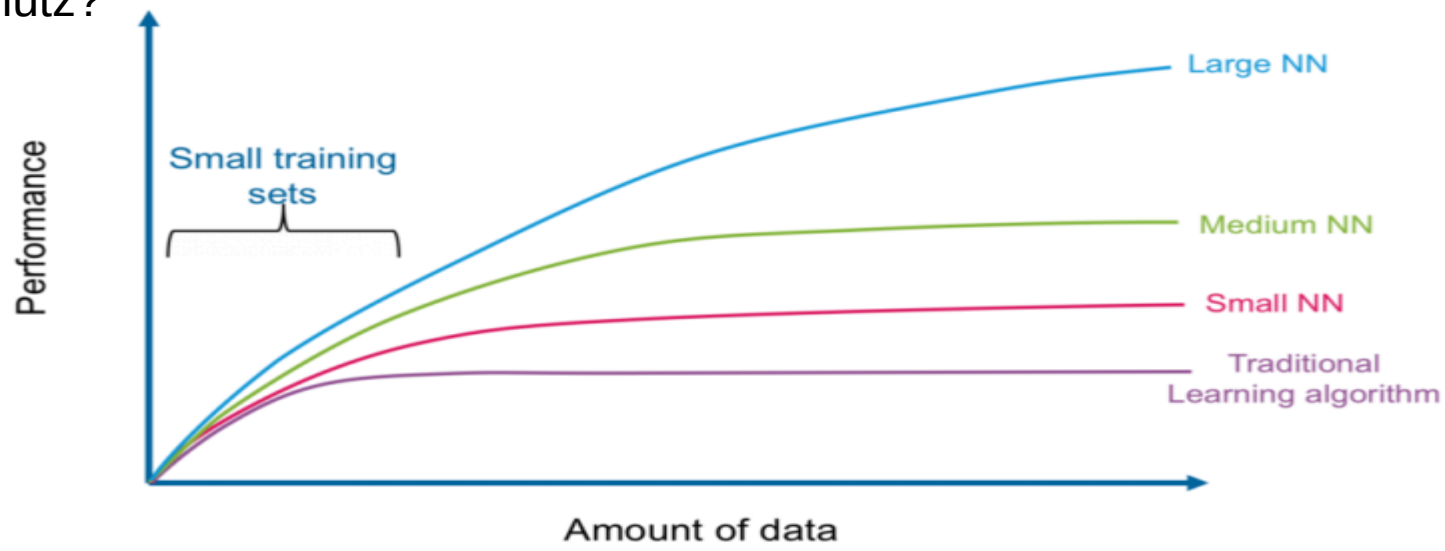
KI-Smart-Story und KI-Melder, Workshop von Frank Stöckel und Uwe Sterr (gestern)

- Aus Sicht des Systemingenieurs ist ML auch nur eine mögliche Lösung.
- ML nicht aus Bequemlichkeit!
- Zumindest für kritische Systeme nicht als alleinige Lösung.

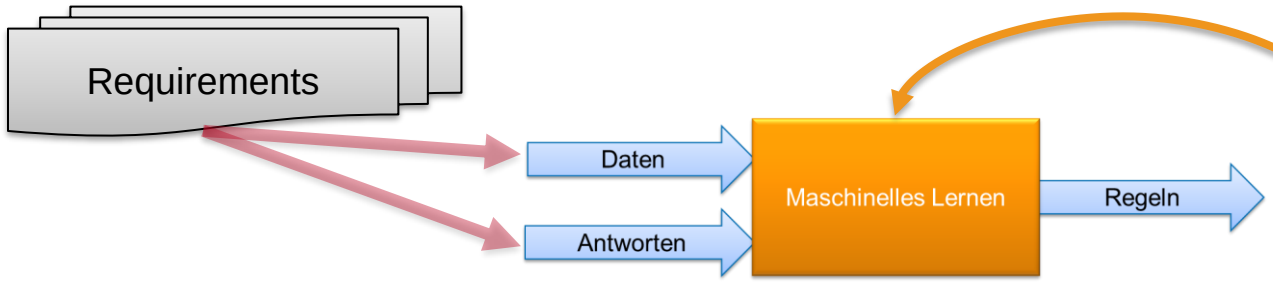


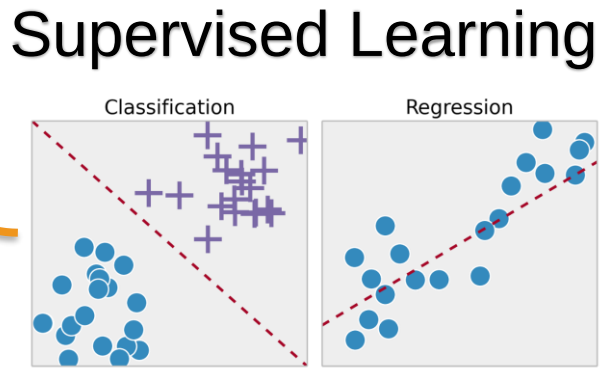
Anforderungen an die Datensammlung

- Der häufigste Grund für das Scheitern von Projekten zur Entwicklung von KI Systemen sind **quantitativ** oder qualitativ **unzureichende Trainingsdaten**.
- Zeit- und Kostenaufwand für das Beschaffen der Trainingsdaten können hoch sein.
- Vorratsdatensammlung lohnt sich!
- Die Daten fallen häufig nicht da an, wo sie benötigt werden:
  - Kunde → OEM → Tier 1 → Tier 2 → Tier x
  - IP-Schutz und Datenschutz?

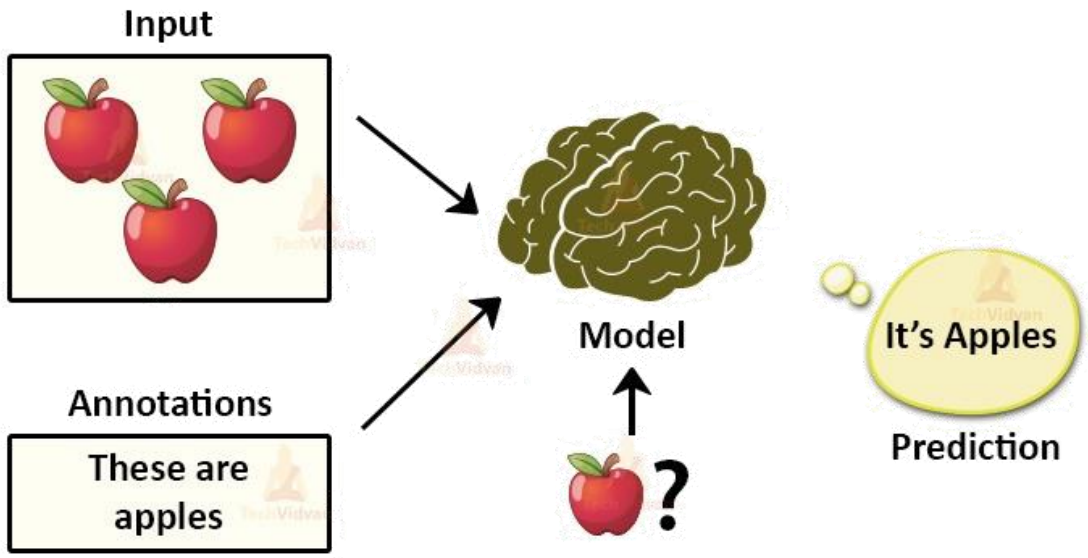


Anforderungen ans Labeling

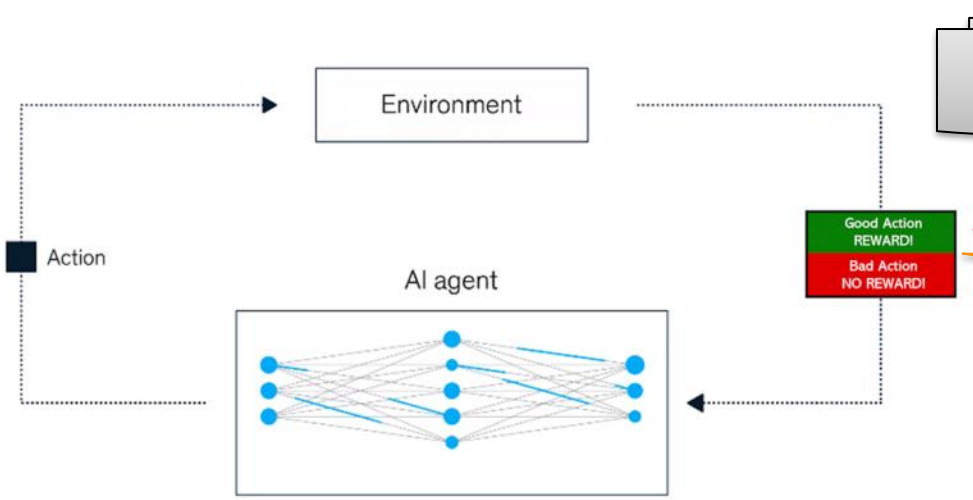


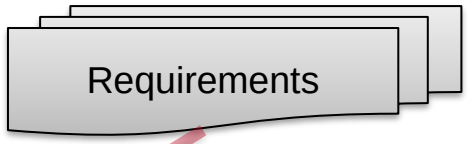


- Wenn die Antworten in den Trainingsdaten falsch sind, wird das System sich dementsprechend verhalten.
- Wer erledigt / verantwortet / prüft heute in der Praxis das Labeling?
- Sollen die Label Wunsch oder Wirklichkeit repräsentieren?



Anforderungen an die Reward-Funktion



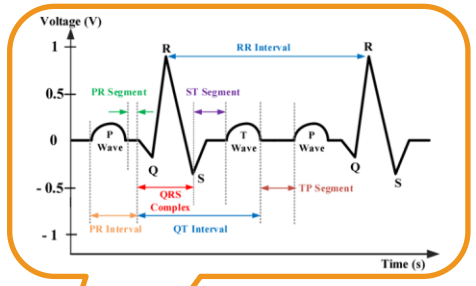


Reinforcement Learning

Reward-Funktion

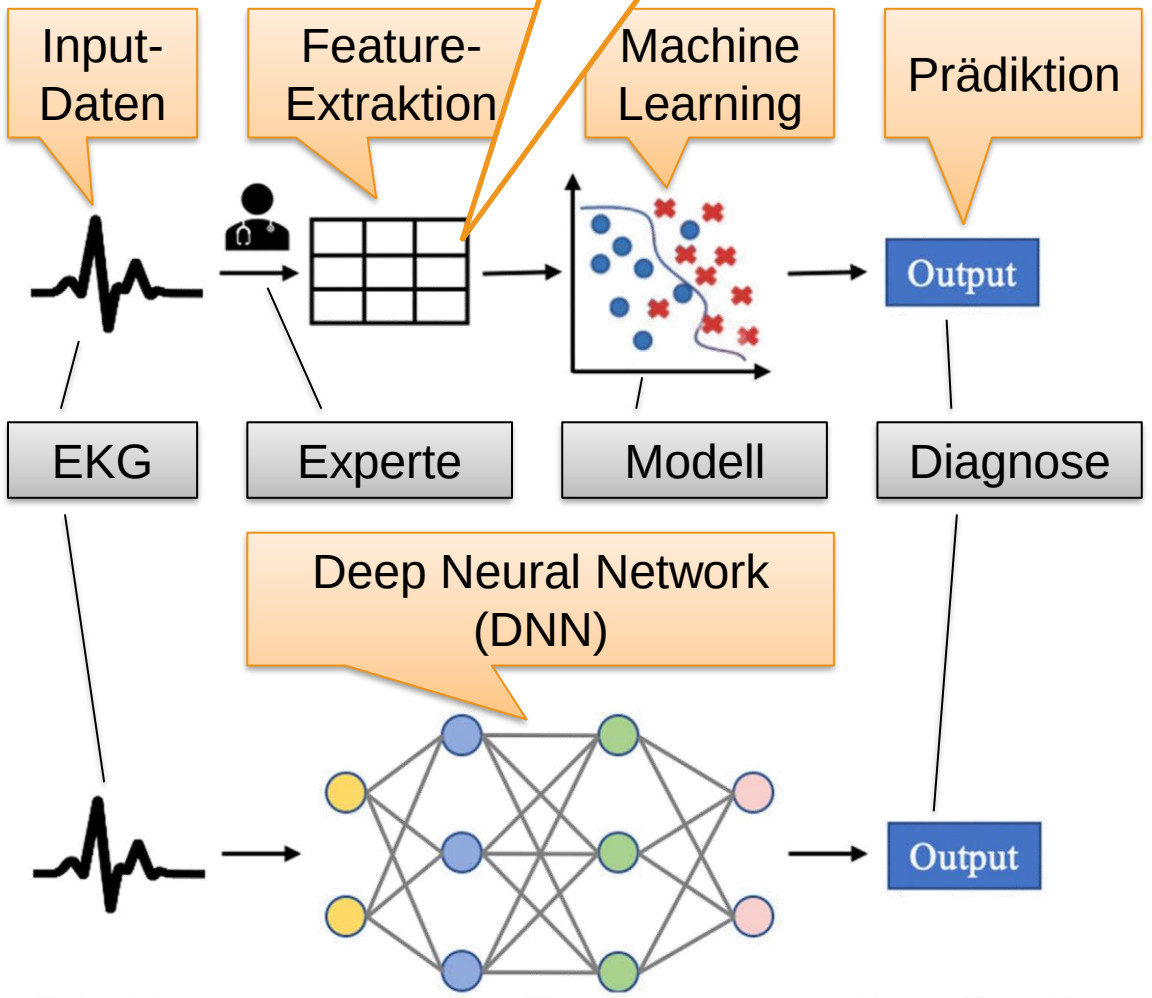
- Die Definition der Reward-Funktion fühlt sich für Entwickler wie klassische Spezifikationsarbeit an…
- **Aber:**
  Die Spezifikation wird nicht von einem mitdenkenden Menschen interpretiert.

Anforderungen an die Feature-Auswahl



„ML Feature“ ≠ „RE Feature“

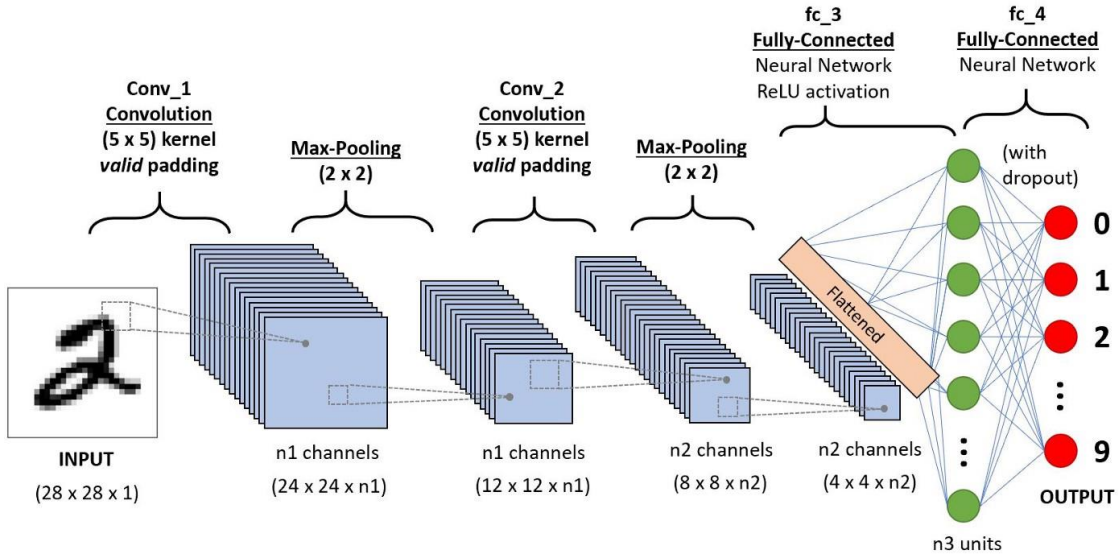
- Über Features können wir gezielt, also Anforderungs-basiert, bestimmen, welche Charakteristika relevant sind.
- Wer macht‘s heute?
  Besser wäre Zusammenarbeit von
  - Domänen-Experten
  - Requirements- / Systems-Engineering
  - ML-Experten / Data scientists
- Beim Deep-Learning lernt das DNN selbst welche Features interessant sind
  - Neue Möglichkeiten
  - Mehr Black-Box, weniger Kontrolle

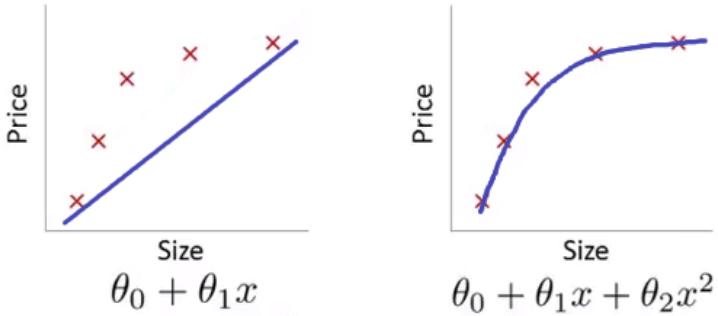


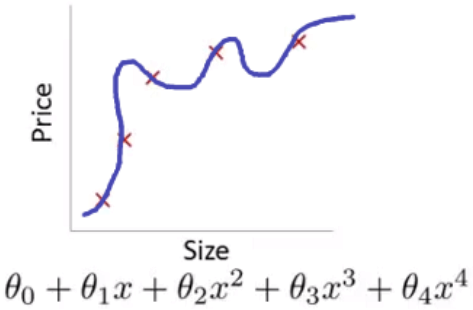
DNNs lernen die Features selbständig

Anforderungen an Modelle und Hyper-Parameter

- Bisher werden Neuronale Netze noch immer hauptsächlich Try-and-Error entwickelt
- Ziel: Gute Performance (Erkennungsrate/F0, Rechenzeit, etc.), insbesondere auch Vermeiden von Over- und Underfitting!
- Hyper-Parameter: Startwerte, Learning-Rate, etc.
  → Auswahl des geeigneten Test-Sets (nicht Validation-Set!)

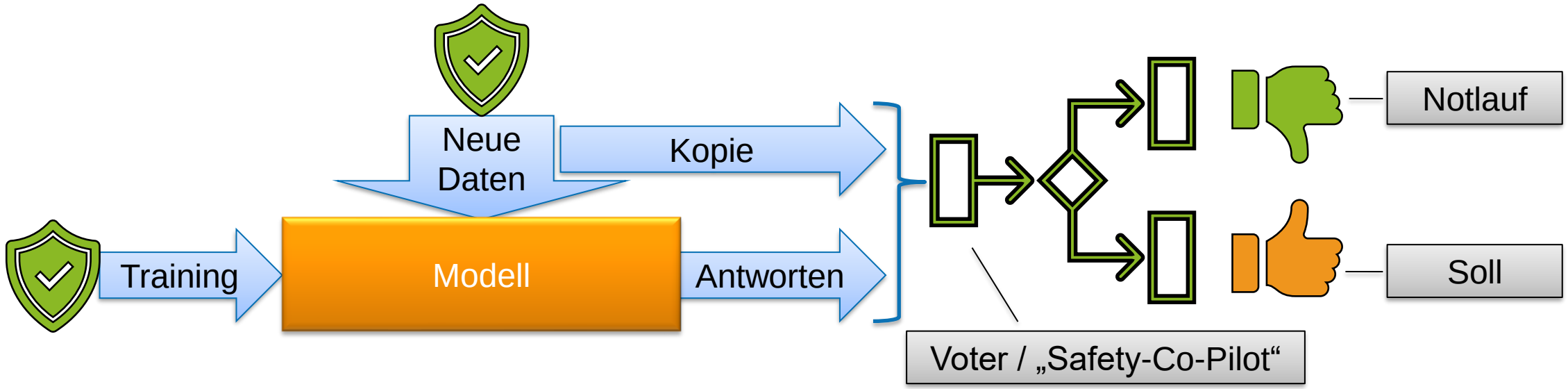






Regelbasierte Überwachung der KI

- Auch wenn die eigentliche Funktionalität nicht mit Regeln beschrieben werden kann, gibt es doch häufig Regeln gegen schlimmes Fehlverhalten.
- Negativ-Requirements erheben!
- Regeln gegen Input-Daten außerhalb des trainierten/erwarteten Bereichs.
- Spezialfall: KI, die im operativen Betrieb oder mit Trainingsdaten des Kunden lernt
  - ➢ Regeln gegen „toxische“ Trainingsdaten/Label

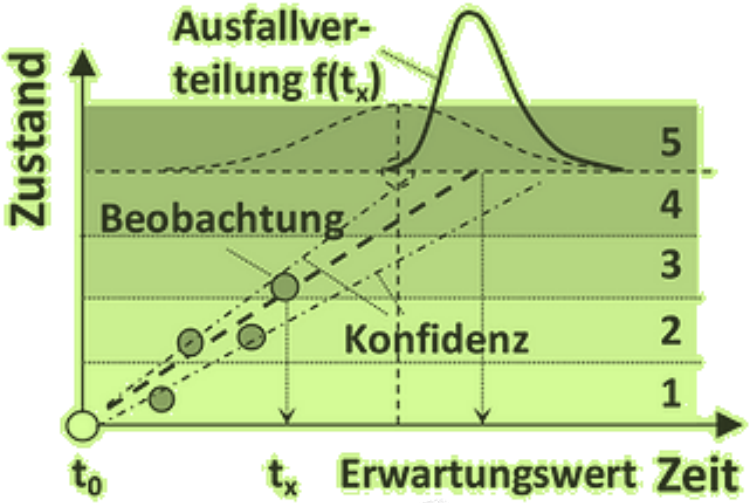


Verifikation und Validierung

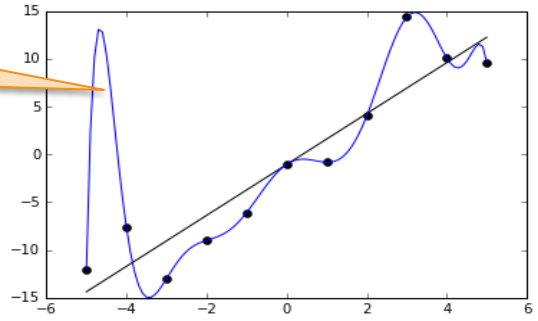
- Akzeptanz-Kriterien für ML sind stochastische Aussagen
  - für Software-Entwickler ein Kulturbruch
  - für System-Ingenieure eigentlich nichts wirklich neues
  - für funktionale Anforderungen aber ungewohnt.

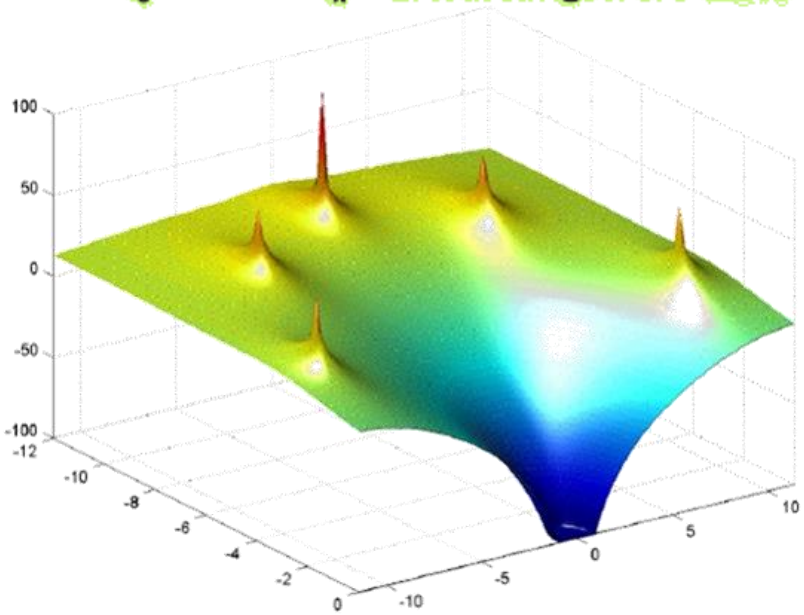
- Validierung der KI-Komponente ≠ Validierung des Systems
  - TDD für ML? Eher nicht möglich.
  - ATDD durchaus.

- Äquivalenzklassenbildung und Grenzwert-Betrachtungen bleiben relevant, reichen aber nicht mehr aus



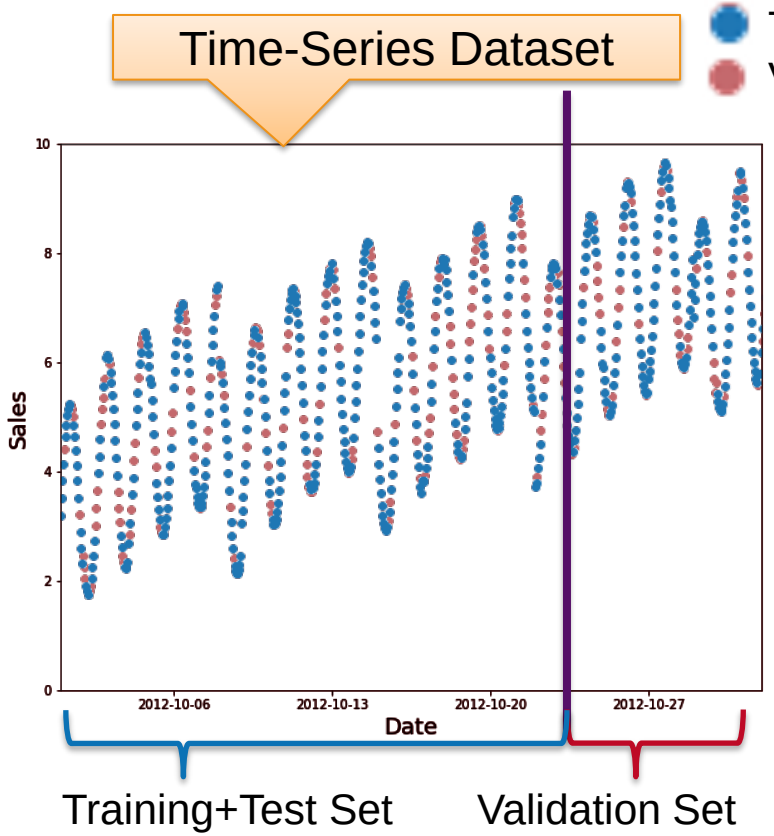
Risiken durch „Overfitting“

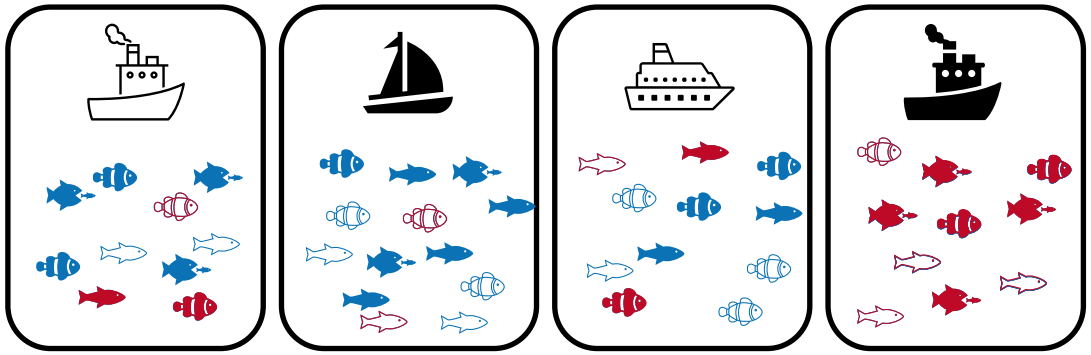




Anforderungen an den Split

- Für eine sinnvolle Auswahl des Validation-Set sind Domänen- und Problem-Verständnis erforderlich.
- Der übliche Zufalls-Split ist häufig keine gute Idee.



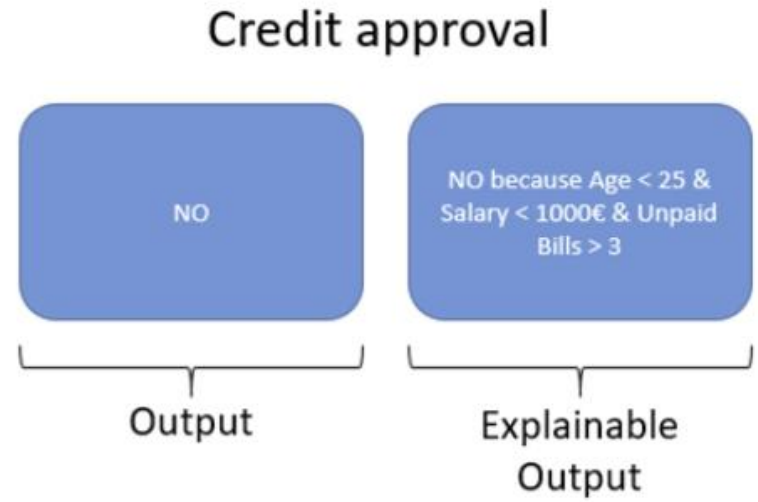


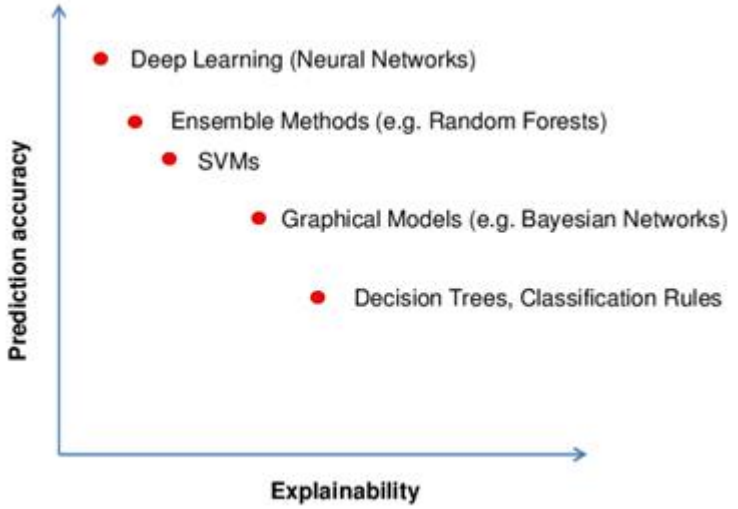
The Nature Conservancy

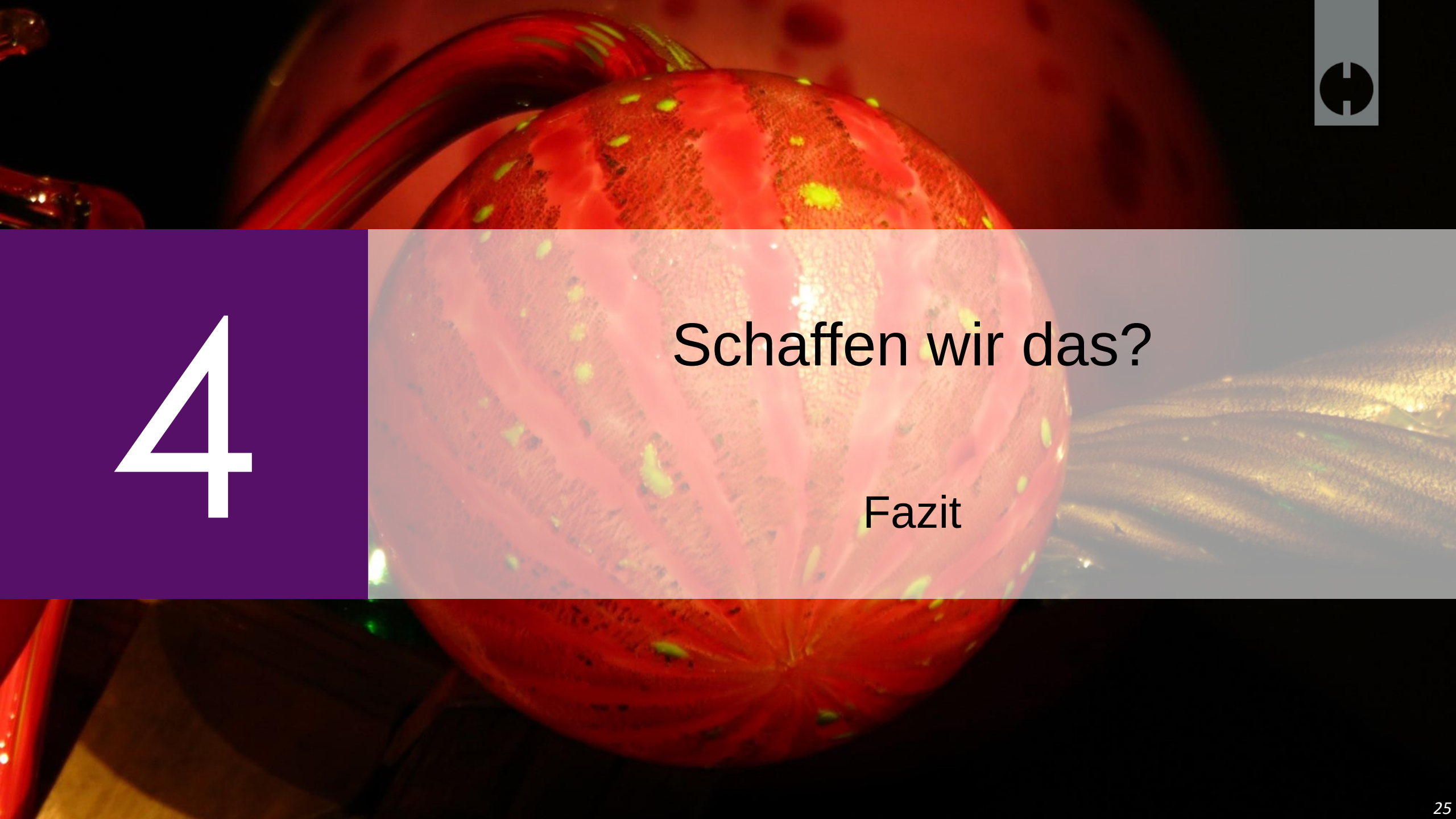
© kaggle

Und was ist mit „Explainable AI?“

- Aktuell großes Forschungsthema
- Wäre regulatorisch sicher erwünscht:
  „Right to Explanation“ (EU)
- Realistisch bleiben!
  - ➢ Je stärker die KI, desto mehr Blackbox!







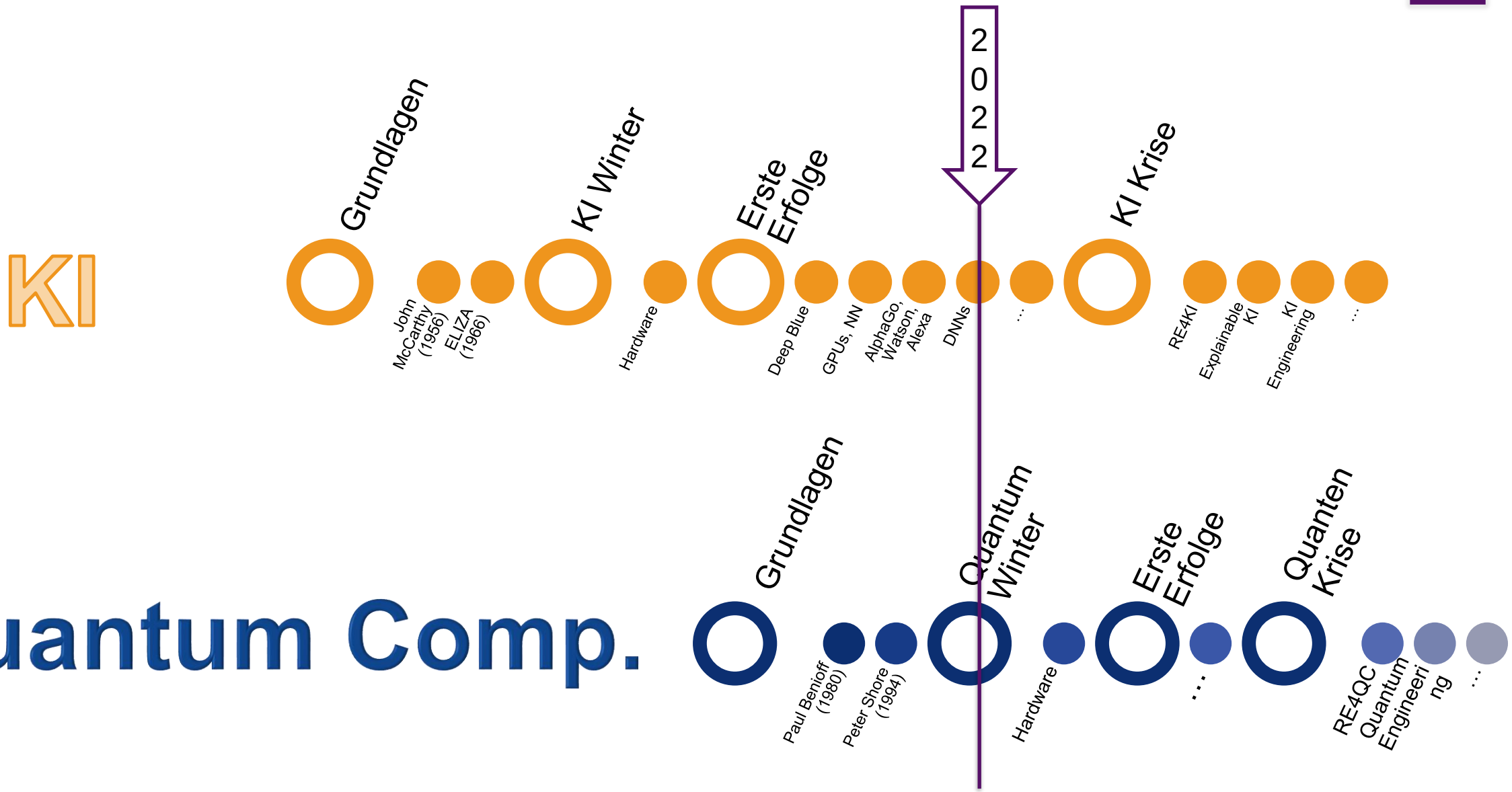
4

Schaffen wir das?

Fazit

Im SW+SE haben wir die Hürden auch genommen

Es warten aber weitere Herausforderungen…



5

Noch Fragen?

jan.ebert@hood-group.com