

“I want AI to do my laundry and dishes so that I can do art and writing,
not for AI to do my art and writing so that I can do my laundry and dishes.”

– “Joanna Maciejewska”

KI in der aktuariellen Produktentwicklung – (wie) geht das ?

Co-Evolution: **Agentic AI** and **Human Agents**

Co-Create Knowledge, orchestrate and operate processes



Versicherungsmathematisches Kolloquium

© msg | Axel Helmert, Thomas Hofmann | 12. Januar 2026



.msg



1 Begrüßung, Überblick und Abgrenzung

10 Min.

- Worum geht es im Vortrag (und was wird nicht behandelt)?
- Wo stehen wir in der Entwicklung der KI in der Branche Insurance?

2 KI in der Branche Insurance

15 Min.

- Welche KI-Methoden sind für welche Einsatzgebiete relevant?
- Was sind die nächsten wichtigen Schritte?

3 Einsatz von Agentic AI in der (aktuariellen-) Produktentwicklung

40 Min.

- Für Neugeschäft, Vertrieb und Verwaltung bestehender Portfolien
- Für die Migration

4 Resümée, Ausblick und Diskussion

25 Min.

KI in der Branche Insurance – From Play to Purpose

1 Wissensmanagement

- Übergang von einfachen Chatbot-Lösungen zu integrierten Assistenten
- Anpassung an domänenspezifische Anforderungen für unterschiedliche Skills
- Quellenbasiertes Arbeiten, schließt Anbindung von Applikationen ein (Jira, Bestandsführung, ...) / Integration im Unternehmen
- Der Nutzer bestimmt den Kontext („Wo suche ich?“), den Grad der Verlässlichkeit, die Tiefe und das Format der Antwort

2 Agentic AI: Automation von Prozessen im Unternehmen

- (Teil-)Autonome KI-Agenten in der Vertragsverwaltung, im Vertrieb und weiteren internen Prozessen
- KI unterstützt den gesamten Softwareengineering Prozess
- Erweiterung und Übertragung der Agentic AI Ansätze auf den Produktentwicklungsprozess
- Integration von Wissensmanagement und Prozessautomation

3 Was sind sinnvolle nächste Schritte?

- **Selbst-lernende Methoden** in domänenspezifischen Umgebungen mit hoher Verlässlichkeit (kein „Training“ oder „Feintuning“)
- Neuro-Symbolische Methoden → Einsatz von generierten **Wissensgraphen** als Grundlage für Wissensmanagement und Prozessautomation
- „**Local Agents**“?

Specialized roles: Generator, Reflector, Curator [1]

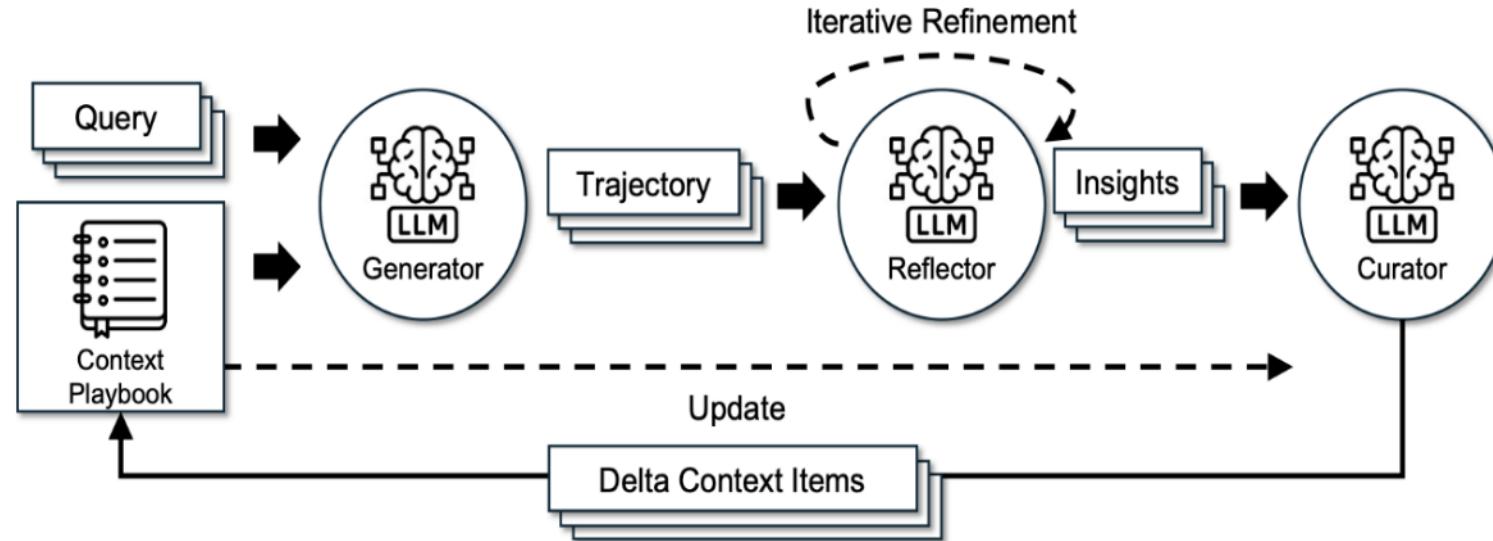


Figure 4: **The ACE Framework.** Inspired by Dynamic Cheatsheet, ACE adopts an agentic architecture with three specialized components: a Generator, a Reflector, and a Curator.

[1] Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models, 6.10.2025, Stanford

1 Agentic Context Engineering (ACE) – Selbstlernende Methoden in domänenspezifischem Kontext

- Kontext-Engineering bezeichnet (teil-)autonome Methoden, die das Verhalten eines Modells verbessern, indem sie additive, natürlichsprachige Informationen für ein LLM erstellen, modifizieren und verwalten, anstatt dessen Gewichte zu verändern.
- Kontexte sind interpretierbar. Sie ermöglichen eine schnelle Integration neuen Wissens und können modellübergreifend geteilt werden.

2 Wissensgraphen / Knowledge Graphs (KGs) [2]

- Die Forschung beschäftigt sich seit Jahrzehnten mit neurosymbolischen Anwendungen, da symbolische Komponenten Abstraktion ermöglichen.
- Die Lösung besteht darin, symbolische und neuronale Ansätze in hybriden, modularen Systemen zu kombinieren.
- Innerhalb eines solchen neurosymbolischen Rahmens können KGs auf natürliche Weise als symbolische Speicher- und Regelrepositorien dienen.
- Besonders relevant in tiefen domänenspezifischen Anwendungen (z.B. der Produktentwicklung).

3 Integration von ACE und KGs in Wissensmanagement und Prozessautomation

- KGs können insbesondere bei der Modellierung im Kontext Produktentwicklung helfen.
- Selbstlernende Methoden beziehen sich auf Mensch-Maschine und Maschine-Maschine Kommunikation

[2] *GraphMERT: Efficient and Scalable Distillation of Reliable Knowledge Graphs from Unstructured Data, Princeton, 10. Oct 2025*

1 Use Agentic AI and **optimize business processes** e2e

- **Automate** to save time, money and human resources.
- Make processes more flexible and **increase customer satisfaction** through new features
- **Hybrid approach:** AI-Agents and human agents work together

2 Consider all relevant **regulatory requirements**

- AI – Act: The use of AI agents does **not lead to classification as a high-risk AI system**.
- **Reasoning models**, transparency, audit trails, and the use of human agents enable compliant design
- In accordance with **Article 22 of the GDPR**.

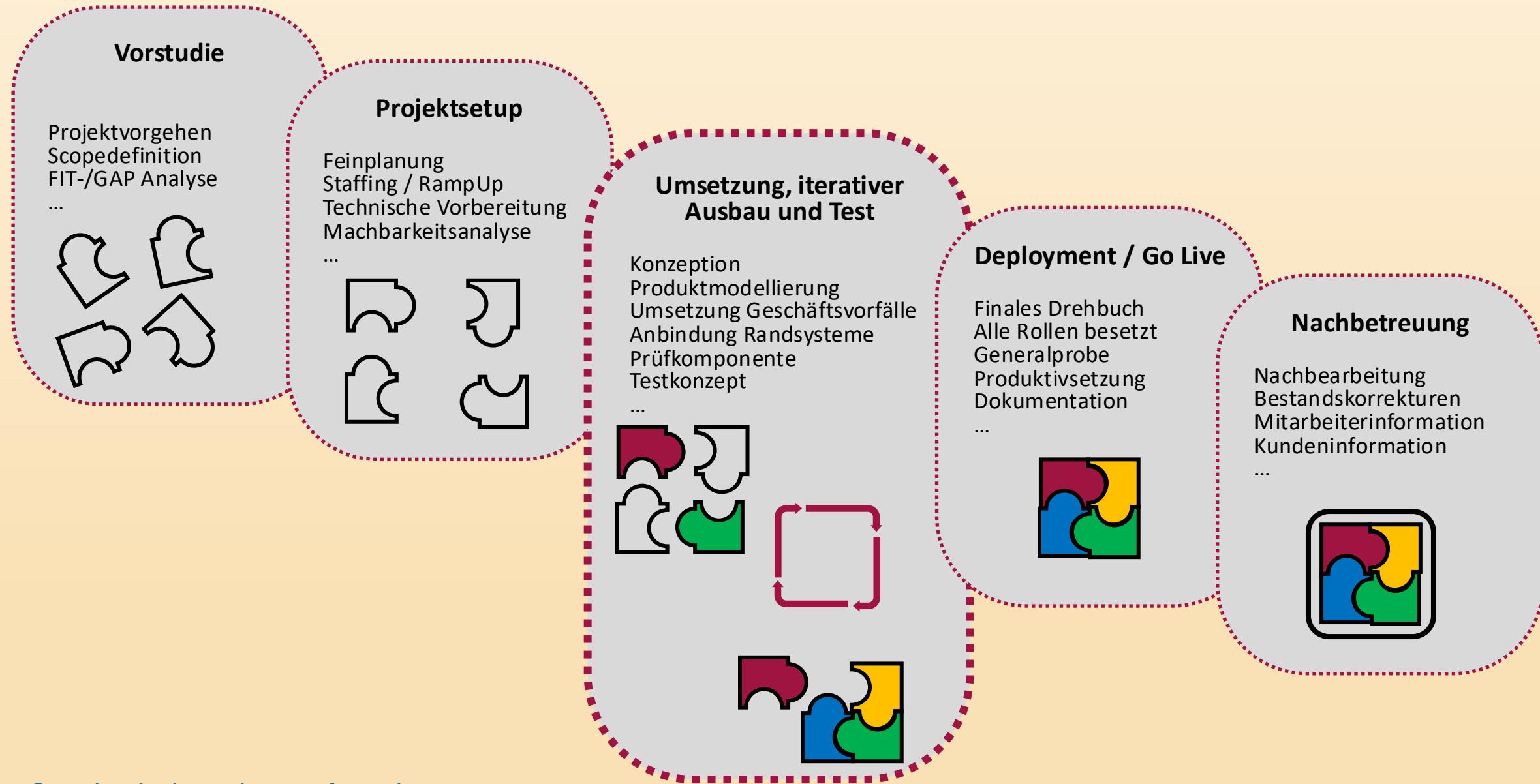
3 Design, Methods and Technology

- Agentic AI: **MCP**, **A2A-Protocols**, **Function calling** and **reasoning models**
- Reuse rule-based processes / **AI agents do not replace core business processes**
- **AI agents bridge gaps** in processes caused by unstructured data.

4 Standardized interfaces (MCP server) to existing communication tools



Produktentwicklungsprozess



Migration vs Neugeschäft

Nicht alles was im Geschäftsplan steht, ist zwingend im Altsystem umgesetzt

- Technische Gründe
- Nachträgliche Änderungen
- (unerkannte) Fehler
- ...

Nicht alles was im Geschäftsplan steht, ist noch relevant

- Falls kein entsprechender (Teil-)Bestand vorhanden ist
- Falls Regeln nicht mehr greifen (Zeiträume/Fristen abgelaufen)
 - Teilauszahler ohne Teilauszahlung
 - alle Verträge schon im beitragsfreien Zustand

Altsysteme enthalten Tarifarten, die nicht mehr angeboten werden

- Heiratsversicherungen
- Kompakttarife
- ...

Alle Geschäftsprozesse müssen unmittelbar nach Migration zur Verfügung stehen

- Beitragsfreistellungen
- Wiederinkraftsetzungen
- Rentenübergänge
- ...

Nicht alles kann im Zielsystem abgebildet werden (z.B. wegen fehlender Daten)

- Es können Konstellationen entstehen, die das Zielsystem nicht erzeugen würde
- Expliziter Test dieser Datenkonstellationen notwendig

Umfangreicher zu migrierender Vertragsbestand!

- Transformation der Vertragsdaten (Datenmodell: Alt- vs Zielsystem)

Produktentwicklungsprozess - KI unterstützt



Dokumente
Bilder
Vertragsdaten
Source-Code
...



Multimodal AI
Machine Learning
SC-Analyse
Stat. Methoden
...

„Wissens“ Extrahierung



Informationen Strukturieren
Vereinheitlichung der Syntax
Komplexitätsreduktion
...

„Wissens“ Strukturierung



Knowledge Hub – Agentic AI

Insurance Domain Knowledge	
Documents	Jira-Tickets
Data models	Databases
Webpages, Wiki	Images, Figures
Source Code	
$k E_x \cdot \left(\overset{erh}{\underset{gg }{E}}(I\ddot{a})_{gg }^{(w)} + v_{erh}^{-gg} \cdot \overset{erh}{\underset{gg }{E}}(I\ddot{a})_{x+k}^{(w)} \right)$	



Zielsystem

„Wissens“ Verwaltung
(Modellierung, Umsetzung und Test)

Mögliche Anwendungsfelder

Vorstudie

- Automatisierte Analyse und Beauskunftung von Versicherungsdokumenten, Bestandsdaten und Quellcode
- Aufbereitung und Strukturierung der Tariflandschaft
- Bestandsanalyse und -bewertung (vor Migrationsbeginn)

Umsetzung, iterativer Ausbau und Test

- Sourcecode-Generierung und Deployment
- Automatisierter Import der Produktdaten und -einstellungen
- Unterstützung des aktuariellen Tests durch einen Zielsystem-unabhängigen Referenzrechner und automatisierte Testfallgenerierung

Deployment und Nachbetreuung

- Informationsarchivierung und -verwaltung in einem *Knowledge Hub*
- Unterstützung im Dokumentationsprozess

Domänen-spezifisches Ablageformat (DSA)

Zielsetzung: erzeuge „*Single source of truth*“

Wird (iterativ) erzeugt aus

- Schriftdokumenten, Bildern, Jira-Tickets,... via LLMs
- Tabulare Vertrags- / Testdaten via Machine Learning Verfahren
- Legacy- / externer Source-Code via SC-Analyse

Basis für die (teil-)automatisierte (Agentic) KI unterstützte Produktentwicklung

```
],
  "rechnungsgrundlagen": {
    "Sterbetafel": [
      "DAV_1994_T",
      "NOT_FOUND",
      "NOT_FOUND",
      "NOT_FOUND"
    ],
    "Zins": 0.0225,
    "Kostensaetze": [
      {
        "name": "\\\alpha^z",
        "value": "40 \\\cdot \\\frac{1}{1000}"
      },
      {
        "name": "\\\alpha^{\\\gamma}",
        "value": "1 \\\cdot \\\frac{1}{1000}"
      },
      {
        "name": "\\\beta",
        "value": "5 \\\cdot \\\frac{1}{100}"
      },
      {
        "name": "\\\gamma_2^T",
        "value": "1.3 \\\cdot \\\frac{1}{1000}"
      }
    ]
  }
}
```

OCR: Optical Character Recognition (Texterkennung)

Arithmetisch fallende Risikolebensversicherung. Die Leistung beginnt in Höhe von $n - 1$ im ersten Versicherungsjahr und fällt auf 0 im letzten Versicherungsjahr

$${}_nAF_x = \frac{(n-1) \cdot M_x - (R_{x+1} - R_{x+n})}{D_x}$$

Geometrisch steigende, unterjährige Leibrente

$${}^{erh}(I\ddot{a})_x^{(w)} = \left(1 - \frac{w-1}{2w} \cdot d_{erh}\right) \cdot \tilde{\ddot{a}}_x - \frac{w-1}{2w} \cdot v_{erh}$$

Temporäre, geometrisch steigende, unterjährige Zeitrente

$${}^{erh}(I\ddot{a})_{\overline{n}}^{(w)} = \left(1 - \frac{w-1}{2w} \cdot d_{erh}\right) \cdot \tilde{\ddot{a}}_{\overline{n}} - \frac{w-1}{2w} \cdot v_{erh} \cdot \left(1 - \tilde{i}E_n\right)$$

Aufgeschobene, geometrisch steigende Leibrente mit Rentengarantiezeit

$${}^{erh}_{kl}(I\ddot{a})_{x:\overline{gg}}^{(w)} = kE_x \cdot \left({}^{erh}(I\ddot{a})_{\overline{gg}}^{(w)} + v_{erh}^{-gg} \cdot {}^{erh}_{gg}(I\ddot{a})_{x+k}^{(w)} \right)$$

Arithmetisch fallende Risikolebensversicherung. Die Leistung beginnt in Höhe von $n - 1$ im ersten Versicherungsjahr und fällt auf 0 im letzten Versicherungsjahr:

$${}_nAF_x = \frac{(n-1) \cdot M_x - (R_{x+1} - R_{x+n})}{D_x}$$

Geometrisch steigende, unterjährige Leibrente:

$${}^{erh}(I\ddot{a})_x^{(w)} = \left(1 - \frac{w-1}{2w} \cdot d_{erh}\right) \cdot \tilde{\ddot{a}}_x - \frac{w-1}{2w} \cdot v_{erh}$$

Temporäre, geometrisch steigende, unterjährige Zeitrente:

$${}^{erh}(I\ddot{a})_{\overline{n}}^{(w)} = \left(1 - \frac{w-1}{2w} \cdot d_{erh}\right) \cdot \tilde{\ddot{a}}_{\overline{n}} - \frac{w-1}{2w} \cdot v_{erh} \cdot \left(1 - \tilde{i}E_n\right)$$

Aufgeschobene, geometrisch steigende Leibrente mit Rentengarantiezeit:

$${}^{erh}_{kl}(I\ddot{a})_{x:\overline{gg}}^{(w)} = kE_x \cdot \left({}^{erh}(I\ddot{a})_{\overline{gg}}^{(w)} + v_{erh}^{-gg} \cdot {}^{erh}_{gg}(I\ddot{a})_{x+k}^{(w)} \right)$$

Multimodal AI

Quality Check

Arithmetisch fallende Risikolebensversicherung. Die Leistung beginnt in Höhe von $\backslash(n-1\backslash)$ im ersten Versicherungsjahr und fällt auf 0 im letzten Versicherungsjahr:
 $\backslash[()_n AF_x = \backslashfrac{(n-1) \cdot \backslashdot{M}_x - (R_{x+1} - R_{x+n})}{D_x}\backslash]$

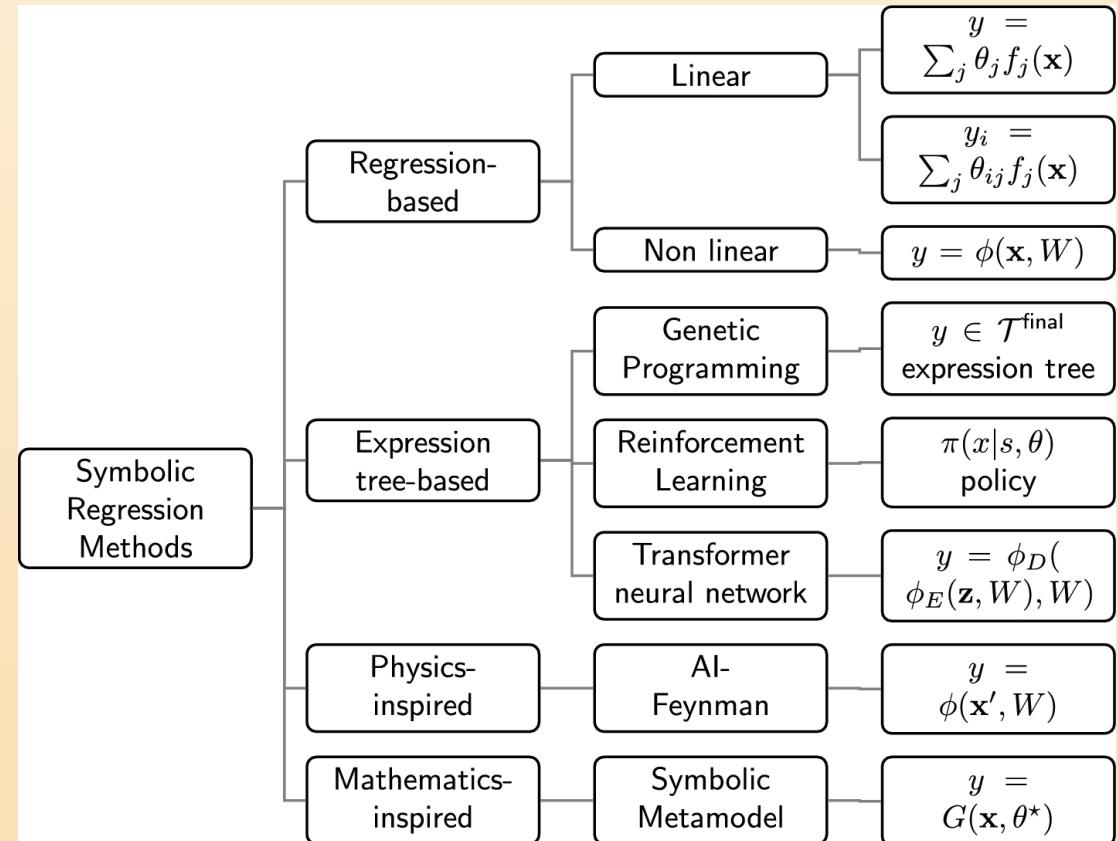
Geometrisch steigende, unterjährige Leibrente:
 $\backslash[()^{erh}(I\backslashdot{a})_x^{(w)} = \backslashleft(1 - \backslashfrac{w-1}{2w} \cdot \backslashdot{d}_{erh}\backslashright) \cdot \backslashdot{\{\tilde{i}\}} \cdot \backslashdot{a}_x - \backslashfrac{w-1}{2w} \cdot \backslashdot{v}_{erh}\backslash]$

Temporäre, geometrisch steigende, unterjährige Zeitrente:
 $\backslash[()^{erh}(I\backslashdot{a})_{\overline{n}}^{(w)} = \backslashleft(1 - \backslashfrac{w-1}{2w} \cdot \backslashdot{d}_{erh}\backslashright) \cdot \backslashdot{\{\tilde{i}\}} \cdot \backslashdot{a}_{\overline{n}} - \backslashfrac{w-1}{2w} \cdot \backslashdot{v}_{erh} \cdot \backslashleft(1 - \backslash{\{\tilde{i}\}} E_n\backslashright)\backslash]$

Aufgeschobene, geometrisch steigende Leibrente mit Rentengarantiezeit:
 $\backslash[()_k(\backslashrule{0.5pt}{1ex})^{erh}(I\backslashdot{a})_{x:\overline{gg}}^{(w)} = \backslash_k E_x \cdot \backslashdot{\{\tilde{i}\}} \cdot \backslashdot{a}_{\overline{gg}} - \backslashdot{v}_{erh}^{-gg} \cdot \backslashdot{a}_{x+k}^{(w)}\backslash]$

Machine Learning: z.B. Symbolische Regression

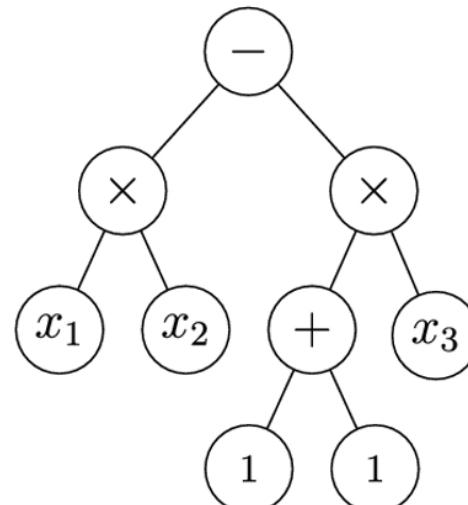
- **Symbolische Regression** ist eine Methode des maschinellen Lernens, mit der man explizit mathematische Formeln aus Daten ermittelt.
- Ziel ist es, eine **lesbare mathematische Formel** zu finden, die die Daten möglichst gut beschreibt.
- Dabei werden die Modellstruktur und die Modellparameter gleichzeitig an die gegebenen Daten angepasst.
- Das Training der Modelle erfolgt dabei rein empirisch.
- Es kann **Vor- bzw. Fachwissen** über den zu modellierenden Zusammenhang berücksichtigt werden, um semi-empirische Modelle zu bilden.



SC-Analyse

- **Static Programm Analysis:** Analysiert Programmcode ohne den Code auszuführen.
- **Dynamic Programm Analysis:** Analysiert Programmcode während der Code ausgeführt wird.
- **Symbolic Programm Analysis:** Arbeitet mit symbolischen anstatt mit konkreten Werten als Programmeingabe.
- **LLMs:** als Pre/Post-processing
 - Semantic Pattern Matching
 - Code-Vervollständigung
 - Automatische Testfallgenerierung
 - ...

Example: $f(x) = x_1 \cdot x_2 - 2x_3$



Paradigm	Core Principle	Primary Artifacts	Code Coverage	Key Strengths	Key Weaknesses	Suitability for Formula Extraction
Static Analysis	Analysis of code structure without execution.	AST, CFG, Data-Flow Facts	All-path (theoretical, often imprecise)	Scalability, speed, early detection in SDLC.	Imprecision (false positives), path/context insensitivity.	Good for local expression identification and dependency analysis.
Dynamic Analysis	Analysis of runtime behavior on concrete inputs.	Execution Traces, Invariants	Partial (dependent on test suite)	Precision on observed paths, handles environment interaction.	Incomplete coverage (false negatives), test suite dependency.	Good for inferring behavior on typical inputs and for complex systems.
Symbolic Execution	Analysis of code logic with symbolic inputs.	Symbolic Expressions, Path Conditions	All-feasible-path (theoretical, often incomplete)	High semantic fidelity, direct generation of formal specifications.	Path explosion, SMT solver limitations, poor scalability.	Excellent for deriving precise, conditional specifications of critical components.