

INNOQ TECHNOLOGY DAY · 12. NOVEMBER 2026

Wenn der Agent die *Schadenspost* sortiert

AI Product Engineering in der Praxis

INNOQ

PETER FRIEDL
NV

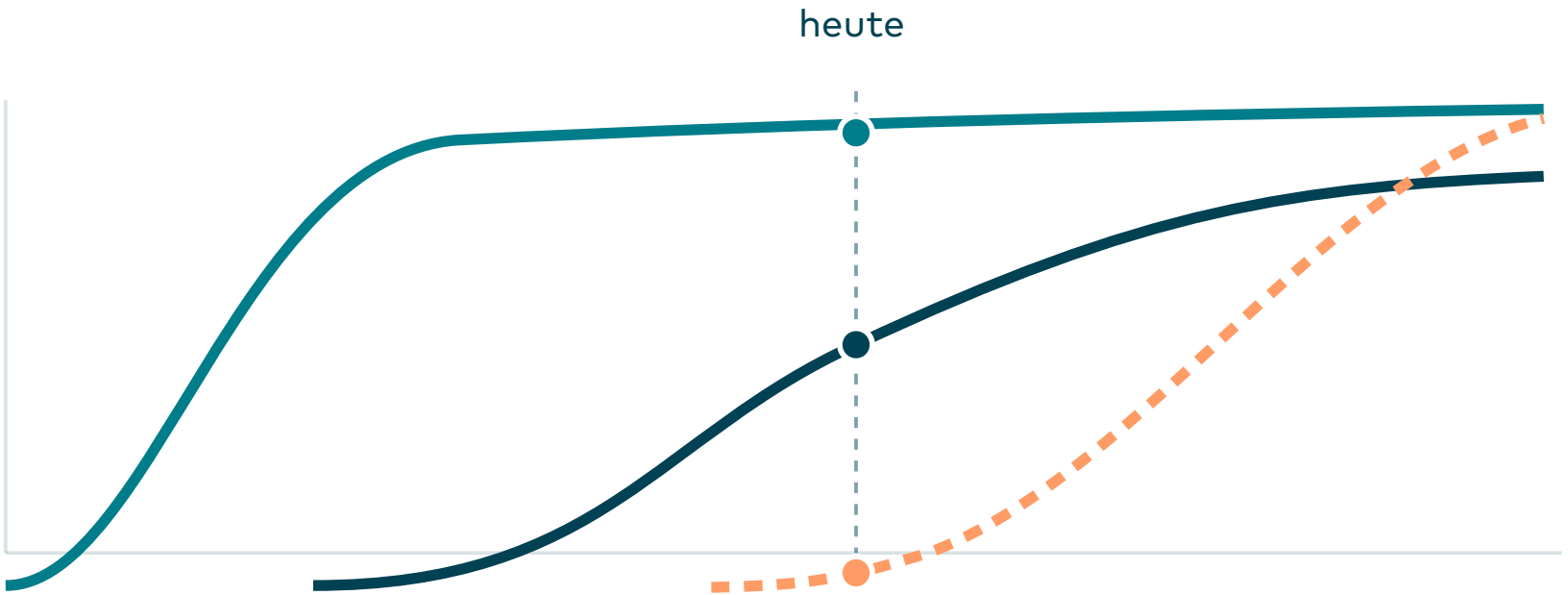
MICHAEL SEEL
SENIOR CONSULTANT

Vier Fragen an euch, Antwort per *Emoji*

❤️ ja 👍 ein bisschen 😮 noch nicht

1. Wer nutzt KI in seiner Arbeit? ChatGPT und Co.
2. Wer nutzt KI, um Software zu entwickeln?
3. Wer hat OpenClaw schon ausprobiert?
4. Wer baut KI-Features in eure eigene Software ein?

AI *Adoption*



Chat · 2023

agentisches Coding · 2025

KI im Kern der Arbeit · 2026

Zwei verschiedene *Fragen*

AGENTIC SOFTWARE ENGINEERING

Wie baue ich Software *mit* KI?

AI PRODUCT ENGINEERING

Wie baue ich Software, die *KI enthält*?

Wer sitzt eigentlich *davor*?

Chatfenster

du liest jede Antwort

Coding Agent

ein Entwickler prüft jeden Commit

Autonomer Agent

niemand



Zwei *Antworten*

Verzichten

zu gefährlich, lassen wir

Einsperren

Druckbehälter, Regelstäbe, Messtechnik


KÖGEL


KÖGEL

Das ging vorher *nicht*

Freitext, emotional

Handyfotos

PDF vom Gutachter

halbe Angaben fehlen

kein Regelwerk kommt da durch

Zwei *Probleme*

Menge

rund 65.000 E-Mails im Jahr

Latenz

bis eine E-Mail beim Sachbearbeiter auf dem Schreibtisch lag

Jede E-Mail muss ins *richtige Team*

E-Mail + PDFs

E-Mail + Fotos

„siehe Anhang“

→

Agent

LLM on-prem + Code

→

- ✓ betrifft einen Schaden?
- ✓ Team und Sparte
- ✓ Nummern und Kennzeichen
- ✓ Dokumente aus den Anhängen
- ✓ Logos und Banner aussortiert

Der Prototyp war schnell *erstaunlich gut*

Wir haben ihn *weggeworfen*.

PROTOTYP	HEUTE
Prompt im Code	Agent Contract, versioniert
Stichproben mit dem Fachbereich	Eval vor jeder Contract-Änderung
jede Datei für sich	multimodales Modell, Dokumente aus mehreren Dateien

„Kann KI das machen?"

LLM

> Schadenfall: Foto vom Unfall, Gutachten, Mietwagenrechnung. Bitte bearbeiten.

● Vielen Dank für Ihre Schadenmeldung. Es tut uns leid, dass Ihr Fahrzeug beschädigt wurde. Wir kümmern uns umgehend darum.

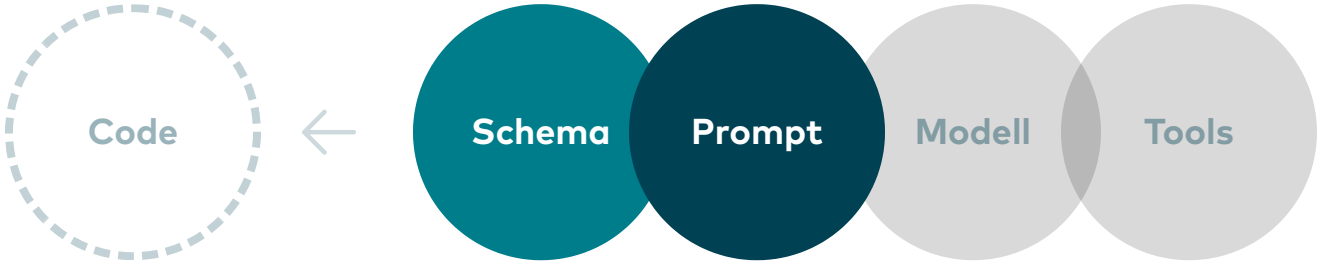
→ **JSON.parse** (antwort)
✗ SyntaxError: Unexpected token 'V'

Struktur *erzwingen*

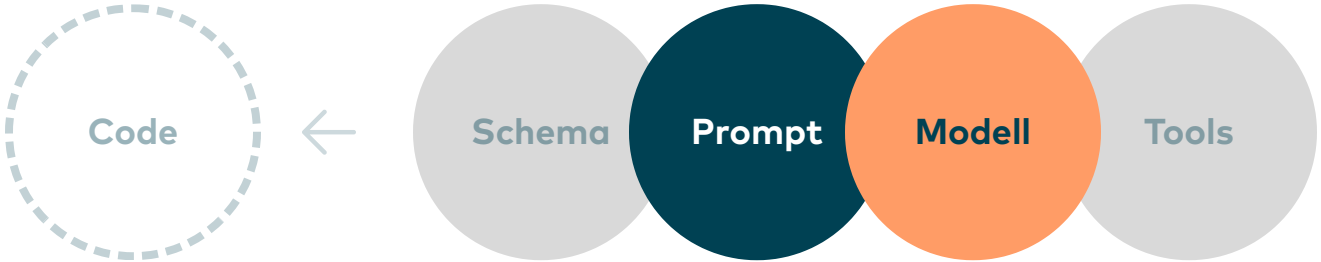
LLM

```
> Antworte als JSON.

● Klar! Hier ist dein JSON: …

→ JSON.parse (antwort)
  ✗ SyntaxError: Unexpected token 'K'

→ response_format schema erzwungen
  {
    "sparte": "Kfz",
    "schadennummer": "KFZ-2026-0042",
    "dokumente": ["Foto", "Gutachten", "Rechnung"]
  }
```

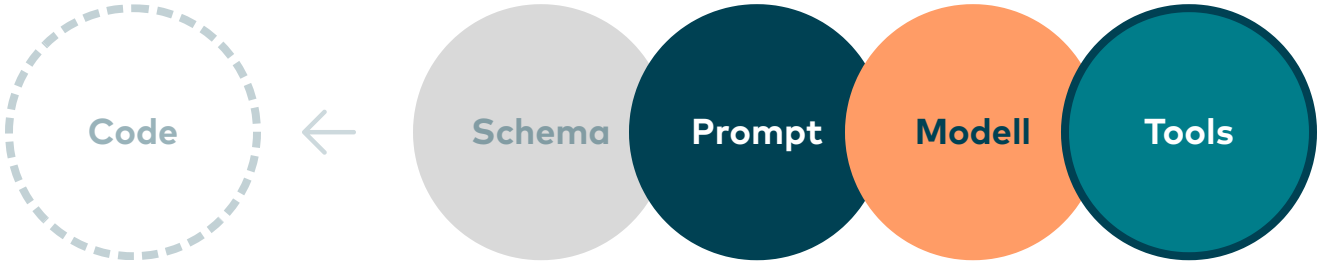
Schema passt zum *Prompt*



Prompt passt zum *Modell*



Tools hängen an Prompt und *Modell*



Durch Tools *handelt* das Modell

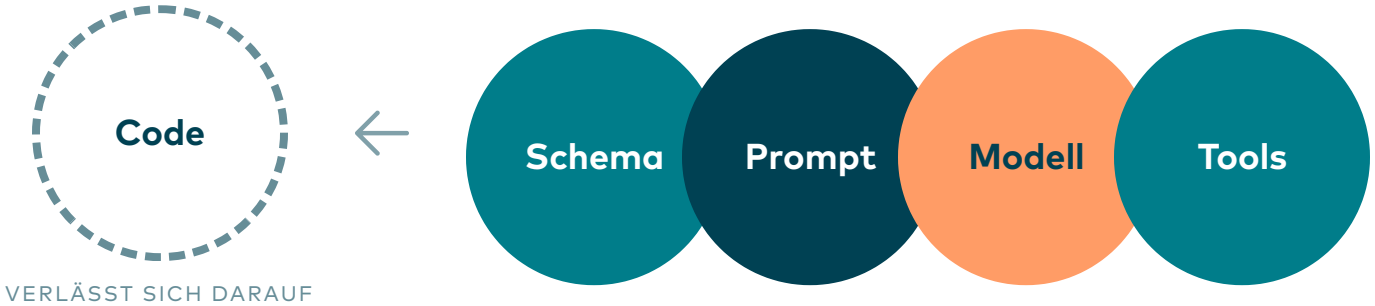
LESEND

Nummer prüfen

SCHREIBEND

6.000 € auszahlen

Der *Agent Contract*

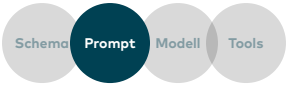


Entwicklung
Schema, Modell, Tools

Fachbereich
Prompt

die gemeinsame Zusicherung von Fachbereich und Entwicklung

Format gültig. Antwort *falsch*.



Ohne *Teambeschreibung* keine zuverlässige Zuordnung

Welche Teams es gibt, weiß der Bounded Context Schaden.

Woran man erkennt, was wohin gehört, steht im Agent Contract.

Die KI kann das auch so. *Meistens.*

FALL	NAHELIEGENDE ANTWORT	WORAUF ES ANKOMMT
Kabinenverglasung am Traktor	Kfz, Glasbruch	**Sparte Feuer**
nur das Scheinwerferglas kaputt	Kasko	**eigenes Team Kleingläser**
Pkw beim Abbiegen von einem Lkw mitgerissen	Kasko	**wer von wem etwas fordert**

Es fehlt vor allem **Hauswissen.**

Was ist eigentlich ein *Schadenfoto?*

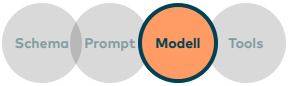
Beschreibung „Foto“

„Foto mit Bezug zum Schaden“

sinngemäß

Foto einer Rechnung: meistens Rechnung, selten Foto.

Hauswissen ist Text, und Text ist *Verhalten*



Ein Default ist eine Entscheidung, die *keiner getroffen* hat

Reasoning Effort

lief unbemerkt auf dem Höchstwert

Temperature

kam ungesetzt vom Server: 1.0

xhigh gegen medium: Team und Sparte im Noise Floor. Seitdem steht medium im Contract.

Kein *Konfidenzwert*

„Wie sicher bist du, dass das zu Kasko gehört?“

Diese Frage kann ein LLM nicht zuverlässig beantworten.

Kein eindeutiges Team? → Fallback-Team. **Ein Mensch entscheidet.**



Das Modell darf nachschlagen, das letzte Wort hat der *Code*

Tool Use

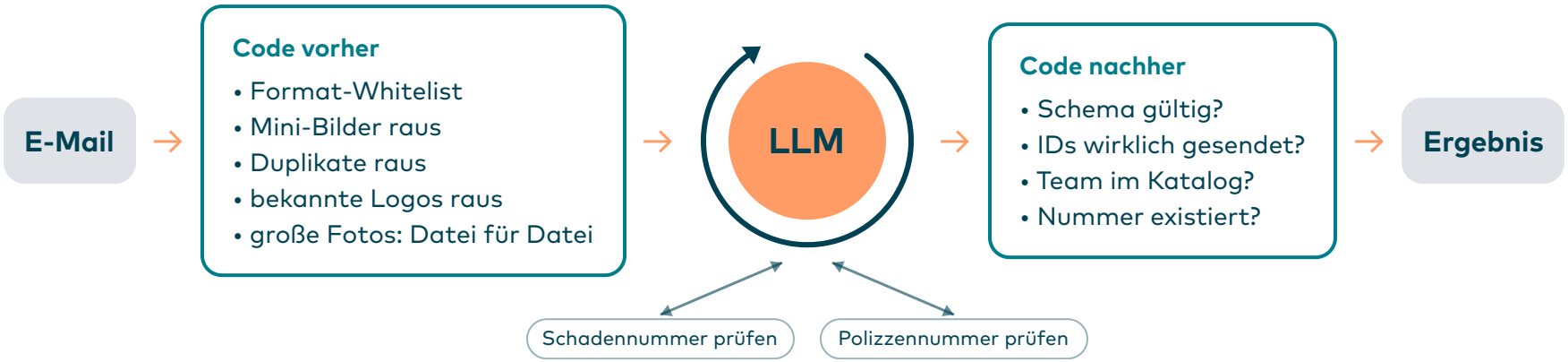
Das Modell darf prüfen, ob eine Schaden- oder Polizzennummer existiert.

Code prüft nach

Jede Nummer prüft der Code selbst im Bestand.

Code ← Schema Prompt Modell Tools

Was ist eigentlich ein *Agent?*



deterministischer Code + probabilistisches Modell = **Agent**

Problem: Was machen wir mit *E-Mails voller großer Fotos?*

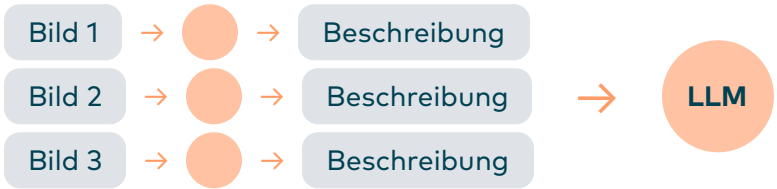
Default: Text und Bilder in einem Call



Multimodal: Das Modell sieht Text und Bilder zusammen und erkennt, dass drei Scans eine Rechnung sind.

Standard, gut 90 % der E-Mails

Ausnahme: Datei für Datei



Bei vielen oder sehr großen Bildern. Jede Datei einzeln beschreiben, dann ein finaler Call.

Preis: Die einzelne Datei sieht den Mailtext nicht.

Problem: Halluzinierte *Datei-IDs*



Gesendet

`eingabeId` 1 2 3

Antwort des Modells

`eingabeId` **4**

passt zum Schema, den Anhang gibt es nicht

Code zur Laufzeit

```
var id = datei.eingabeId();

if (id != null && !gesendeteEingabeIds.contains(id)) {
    throw new IllegalStateException(
        "keine gesendete Eingabe-Id");
}
```

Nur gesendete IDs kommen durch. Sonst: **Repair-Call, dann Retry.**

Code ← Schema Prompt Modell Tools

Problem: Der leere *Erfolg*

Damals

E-Mail + PDF → Aufbereitung (Decoder fehlt) → **LLM** fast weiße Seite → „irrelevant“ → ✓ **Erfolg**

Heute

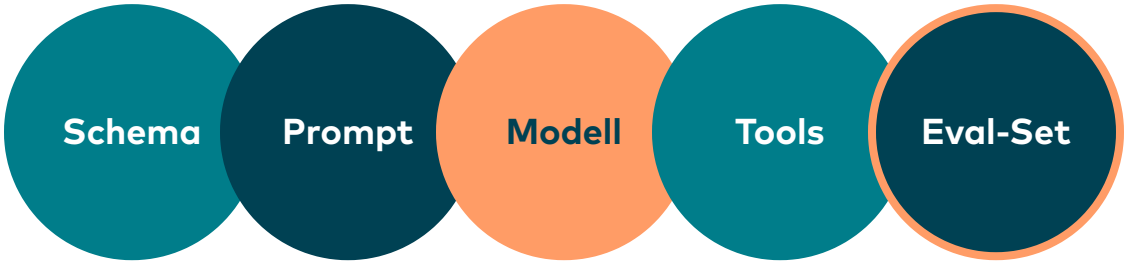
Aufbereitung (Decoder mitgeliefert) → Bildformat unlesbar? → **sichtbar unter „Nicht verarbeitet“**

Wie testet man *Nicht-Determinismus*?

Die *Eval-Schleife*



Der Contract *wächst*



Jede Version des Contracts läuft gegen dasselbe Set.

Keine Prompt-Änderung ohne *Eval*

- Eval-Run vorher und nachher, bei jeder Änderung am Contract
- auch bei Modell-, Engine- und Parameterwechsel
- und vor dem Umschalten vom bisherigen System

Ein Eval-Set, das wie der Betrieb läuft

Ein Eval-Set ohne *Kundendaten*

synthetisch

erfundene E-Mails für seltene Fälle

anonymisiert

echte E-Mails, mit dem eigenen Modell im Haus umgeschrieben

Fotos

Schadenfotos ohne Personenbezug

Jedes Bild sichtet ein Mensch. Die richtigen Antworten legt die Schadenabteilung fest.

Welcher Fehler ist *teurer*?

Ist dieser Anhang relevant?

MITGESCHLEPPT

Ein Logo landet beim Sachbearbeiter

er schaut kurz hin und ignoriert es



AUSSORTIERT

Ein Schadenfoto wird als irrelevant aussortiert

fehlt als eigenes Dokument in der Akte

Wie wir *messen*

je Feld

Team, Sparte, Dokumentart: richtig oder falsch.

asymmetrisch

Ein aussortiertes Schadenfoto wiegt im Betrieb schwerer als ein mitgeschlepptes Logo.

im Kontext

Wer stets „Schaden“ sagt, liegt schon zu 92 % richtig

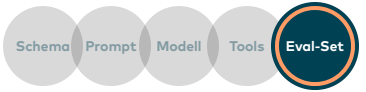
Erst den *Noise Floor* messen

12 / 989

Felder, die sich zwischen zwei identischen Runs unterscheiden

Eine Verbesserung, die kleiner ist als der Noise Floor, ist nicht belegt.

Grenzfälle laufen auf beiden Ständen je fünfmal.



Unser größter Qualitätssprung war ein *Bug im Eval*

75 % → 98 %

Trefferquote beim Team, sobald das Eval den Mailtext mitgab, den die Produktion schon immer hatte

Ein Eval, das das Modell anders füttert als die Produktion, misst nur sich selbst.

Dieselbe Lehre gilt für den Weg durchs Modell: Heute läuft das Eval wie der Betrieb

Lernen ja, *Auto-Lernen* nein



Jede Lektion landet versioniert: im Contract, als Regel im Code oder als Fall im Eval-Set.

z. B. Sturm, Kfz, kunde@… · Strg+K

STATUS: Alle | TEAM: Alle | SCHADENS…: Alle | BEWERTUNG: Alle

5 Elemente verfügbar

✓ Verarbeitet 29.10.2026 18:48
Rückfrage zu meinem Schaden
moritz.muster@example.org
Schaden-Fallback

✓ Verarbeitet 29.10.2026 18:41
Newsletter: Herbstaktion
newsletter@autohaus-beispiel.example
Archivierung

✓ Verarbeitet 29.10.2026 18:34
Rechnung Tausch Windschutzscheibe
buchhaltung@glaserei-muster.example
Kasko-Team › Kasko

✓ Verarbeitet 29.10.2026 18:27
Sturmschaden am Dach
erika.beispiel@example.org
Gebäude-Team › Sturm

✓ Verarbeitet 29.10.2026 18:20
Schadenmeldung: Pkw von Lkw beim Abbi…
max.mustermann@example.org
Kasko-Team › Kasko

Poststück 5 von 5

Live · aktualisiert alle 5s

01a188d2-47c8-71e2-8289-7bf7693bee2e

✓ Verarbeitet | Originale E-Mail (.eml)

KI-Zusammenfassung
Geschädigter meldet, dass sein Pkw beim Abbiegen von einem Lkw mitgerissen wurde; Front rechts beschädigt. Fotos, Kostenvoranschlag der Werkstatt und Unfallbericht liegen bei.

Details | Dokumente 5 | Irrelevante Anhänge 1 | Nicht verarbeitet 0

SCHADENRELEVANT	ja	TEAM	Kasko-Team
SCHADENSPARTE	Kasko		

Stimmt die Zuweisung?

Dein Feedback: Fremdverschulden: Der Lkw ist bei uns versichert, der Pkw-Lenker ist Geschädigter. Gehört zur Kfz-Haftpflicht, nicht zur Kasko.

Echte Post, echte *Bewertung*

95 %

228 von 241

E-Mails beim richtigen Team, inklusive Zustellung über die verifizierte Schadennummer

2 %

5 von 241

E-Mails mit inhaltlich falschem Team

96 %

783 von 817

Dokumente mit der richtigen Dokumentart

Von der Schadenabteilung bewertet, Fälle selbst gewählt, 22.09. bis 08.10.2026, Test-Stage mit echter Post.

Was beim KI-Produkt geholfen hat, was wir *verworfen* haben

geholfen

- ✓ Agent Contract mit Hauswissen
- ✓ das letzte Wort hat der Code
- ✓ Eval vor jeder Contract-Änderung

verworfen

- × OCR-Pipeline vor dem Modell
- × Teambeschreibungen beim Nachbarsystem
- × Konfidenzwert im Schema

Nächstes Mal *anders*

- Das Eval-Set vor dem ersten Prompt bauen
- Alle Modellparameter ab Tag 1 in den Contract, auch die, die man nicht anfasst
- Das Eval füttert das Modell wie die Produktion: gleiche Eingabe, gleicher Weg durchs Modell
- Zuerst den Noise Floor messen, dann die Verbesserung

Ein großer Agent oder viele *kleine*?

ein Agent

Mail lesen | Fall anlegen | Unterlagen anfordern | Deckung prüfen | Schaden bewerten | auszahlen

alle Tools, alle Rechte

Posteingangs-Agent

↓

Schadenfall-anlegen-Agent

↓

Schaden-prüfen-Agent

↓

Schaden-bearbeiten-Agent

je eigene Rechte, je eigener Ausschnitt

Fazit

Der Prompt war das *kleinste* Problem

Contract, Code drumherum, Evals und Betrieb: Dort entscheidet sich der Weg in die Produktion.


WITH GREAT POWER
COMES GREAT RESPONSIBILITY

Danke! Fragen?



PETER FRIEDL
NV



MICHAEL SEEL
SENIOR CONSULTANT

INNOQ

Zum *Weiterlesen*

MICHAEL SEEL · MICHAELSEEL.COM

Der Agent Contract

AI Product Engineering: Qualitätsmerkmale

Die Taube sieht besser

Agent after Agent

Agent-First Design

RUBEN VITT · RUBEEN.DEV

Evals in die CI

Vom Incident zum Regressionstest

Tools sind keine Prompts

Approval ist kein UX-Bug

Observability für AI-Features

Release-Disziplin für AI-Features

Revisionssicheres Logging