

Agentic Coding jenseits des Hypes

Was CTOs, CEOs und Investoren jetzt wissen müssen



Andreas Heubl
BayBG Venture Capital



Veronika von Dehn
BayBG Venture Capital



Dr. Florian Deibenböck
CQSE



Dr. Benjamin Hummel
CQSE

Schlagzeilen 2026 zwischen Hype und Ernüchterung

Prognosen

„[...] we might be six to 12 months away from when the model is doing most, maybe all of what software engineers do end to end.“ [1]

Dario Amodei, CEO Anthropic

WEF Davos, Januar 2026

Neue Kennzahlen

„Today, 75% of all new code at Google is now AI-generated and approved by engineers [...]“ [2]

Sundar Pichai, CEO Google

Google Blog, 22. April 2026

Sicherheitsrisiken

„Autonomous, AI-driven offensive tooling is no longer theoretical.“ [3]

Hugging Face

Security incident disclosure, 16. Juli 2026

Laut OpenAI drangen Modelle aus einer Testumgebung bis in die Produktivinfrastruktur von Hugging Face vor. [4]

Quellen:

[1] World Economic Forum, Radio Davos: „The day after AGI: Two ‘rock stars’ of AI on what it will mean for humanity“, Annual Meeting 2026: <https://www.weforum.org/podcasts/radio-davos/episodes/ai-agi-dario-amodei-demis-hassabis/>

[2] Google Blog, Sundar Pichai: „Cloud Next ‘26: Momentum and innovation at Google scale“, 22.04.2026: <https://blog.google/innovation-and-ai/infrastructure-and-cloud/google-cloud/cloud-next-2026-sundar-pichai/>

[3] Hugging Face: „Security incident disclosure — July 2026“, 16.07.2026: <https://huggingface.co/blog/security-incident-july-2026>

[4] OpenAI: „OpenAI and Hugging Face partner to address security incident during model evaluation“, 21.07.2026: <https://openai.com/index/hugging-face-model-evaluation-security-incident/>

BayBG Venture Capital - BACKING BAVARIA'S BEST STARTUPS

25+

Jahre Venture Capital

80+

Startups finanziert

~200 Mio. €

in Startups investiert

1–10 Mio. €

pro Startup inkl. Folgerunden



Evergreen statt Fonds

Keine Fondslaufzeit – wir begleiten langfristig und bedarfsgerecht.



Flexible Rolle

Lead, Co-Lead, Co-Investor oder Spiegelpartner öffentlicher Investoren.



Klarer Tech-Fokus

Software und Deep Tech, von Late Seed bis Series B.



Andreas Heubl

Senior Investment Manager



Veronika von Dehn

Investment Manager

Time to say goodbye ?

... zum SaaS-Playbook



- ✗ Entwicklung ist der Engpass
- ✗ Für jedes Problem ein SaaS-Abo
- ✗ Frühe Finanzierungsrunden finanzieren Entwickler
- ✗ Feature-Vorsprung schützt
- ✗ Hohe Multiples für ARR

Warum uns Agentic Coding umtreibt



Kapitaleffizienz

Kleinere Teams liefern mehr. Burn Rate, Kapitalbedarf und Rundengrößen verschieben sich.

Welcher Produktivitätsgewinn ist real?



Tempo wird Commodity

Wenn alle schneller bauen, ist Geschwindigkeit allein kein Vorsprung mehr.

Wo liegt künftig der Moat?



Code als Risiko im Asset

Code ist der Kern des Unternehmenswerts. Qualität, Technical Debt, Security, Lizenzen.

Wer verantwortet Agenten-Code?



Due Diligence neu denken

„Wir nutzen KI“ steht in jedem Pitch Deck. Die Aussage allein trägt nicht.

Woran erkennen wir Engineering-Reife?

Ein Blick in den Code



2009

Spin-off der TU München

60+

Mitarbeitende

23

Promotionen in Software-
Engineering

300+

Unternehmen nutzen Teamscale



Dr. Florian Deißeböck

Gründer & Geschäftsführer



Dr. Benjamin Hummel

Gründer

Hype und Kritik

Reifegradmodell für Agentic Software Engineering

Wo Unternehmen heute wirklich stehen

Praxiserfahrungen aus unserer eigenen Entwicklung

Qualität, Governance und Verantwortlichkeit

Ausblick & Takeaways



SIEMENS



Dolby

SW//M

—EnBW

...



Allianz



SEW
EURODRIVE



Vibracoustic

...



Professional Services

Quality Control · Test Intelligence · Software Audits · Due Diligence · IT-Strategie-Beratung



Eigene Software-Entwicklung

Qualitätsanalyse-Plattform Teamscale · 20 Personen Dev-Team · 2 MLOC · SaaS & On-Prem



Akademischer Hintergrund

23 Promotionen · Promotionsprogramm · Eigene Forschung · Enger Kontakt zu Universitäten

► Hype und Kritik

Reifegradmodell für Agentic Software Engineering

Wo Unternehmen heute wirklich stehen

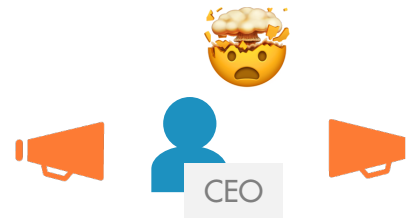
Praxiserfahrungen aus unserer eigenen Entwicklung

Qualität, Governance und Verantwortlichkeit

Ausblick & Takeaways

»\$260 million and 4,500 developer-years saved.«
Andy Jassy, Amazon

»We're not adding any more software engineers next year.«
Marc Benioff, Salesforce



» AI adoption reduced delivery throughput...«
Google DORA

»AI can't replace software engineers yet.«
Marc Benioff, Salesforce



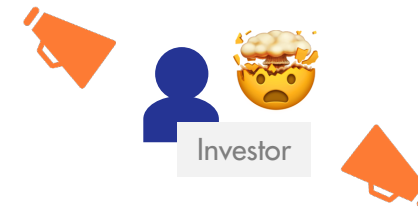
»AI is writing 90 percent of the code.«
Dario Amodei, Anthropic

»Most of the code will be written by AI.«
Mark Zuckerberg, Meta



»SaaSocalypse: \$285 billion wiped from software stocks in 48 hours.«
Jefferies

»Software once ate the world. Now it is in danger of eating itself.«
The Economist



»AI made experienced developers 19% slower.«
METR Study

»Users with an AI assistant wrote less secure code.«
Neil Perry, Stanford

»History and economics say otherwise.«
Fortune

»Some corners of the SaaS industry are booming thanks to AI.«
The Economist

Hype und Kritik

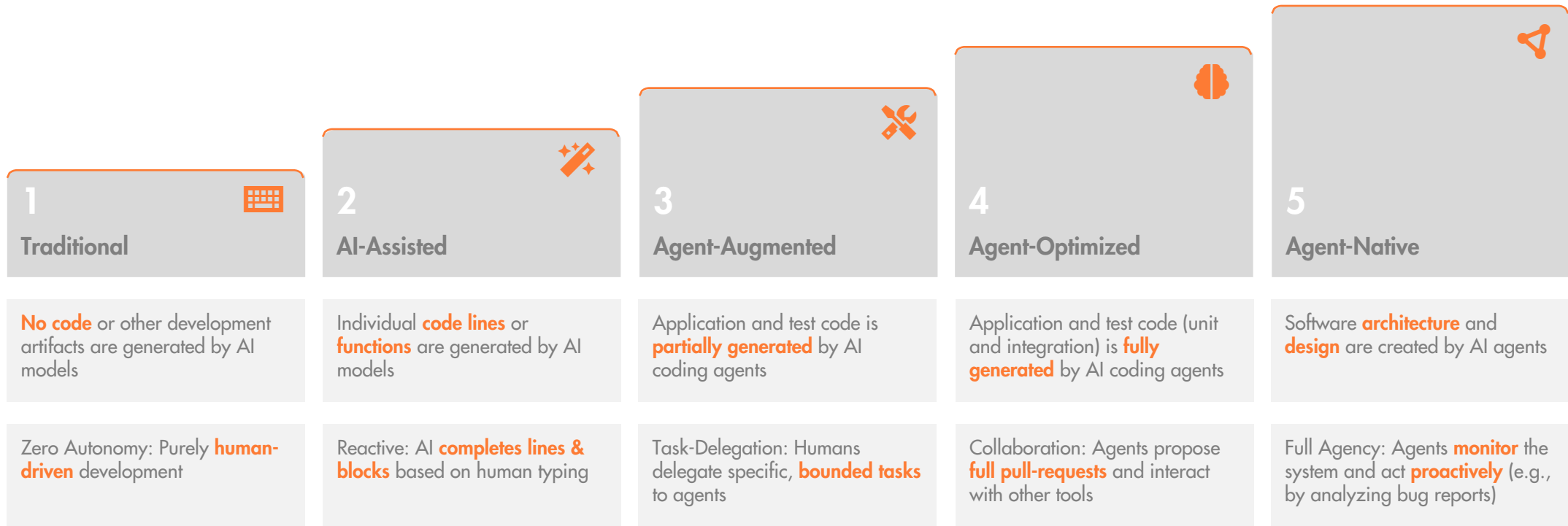
▶ **Reifegradmodell für Agentic Software Engineering**

Wo Unternehmen heute wirklich stehen

Praxiserfahrungen aus unserer eigenen Entwicklung

Qualität, Governance und Verantwortlichkeit

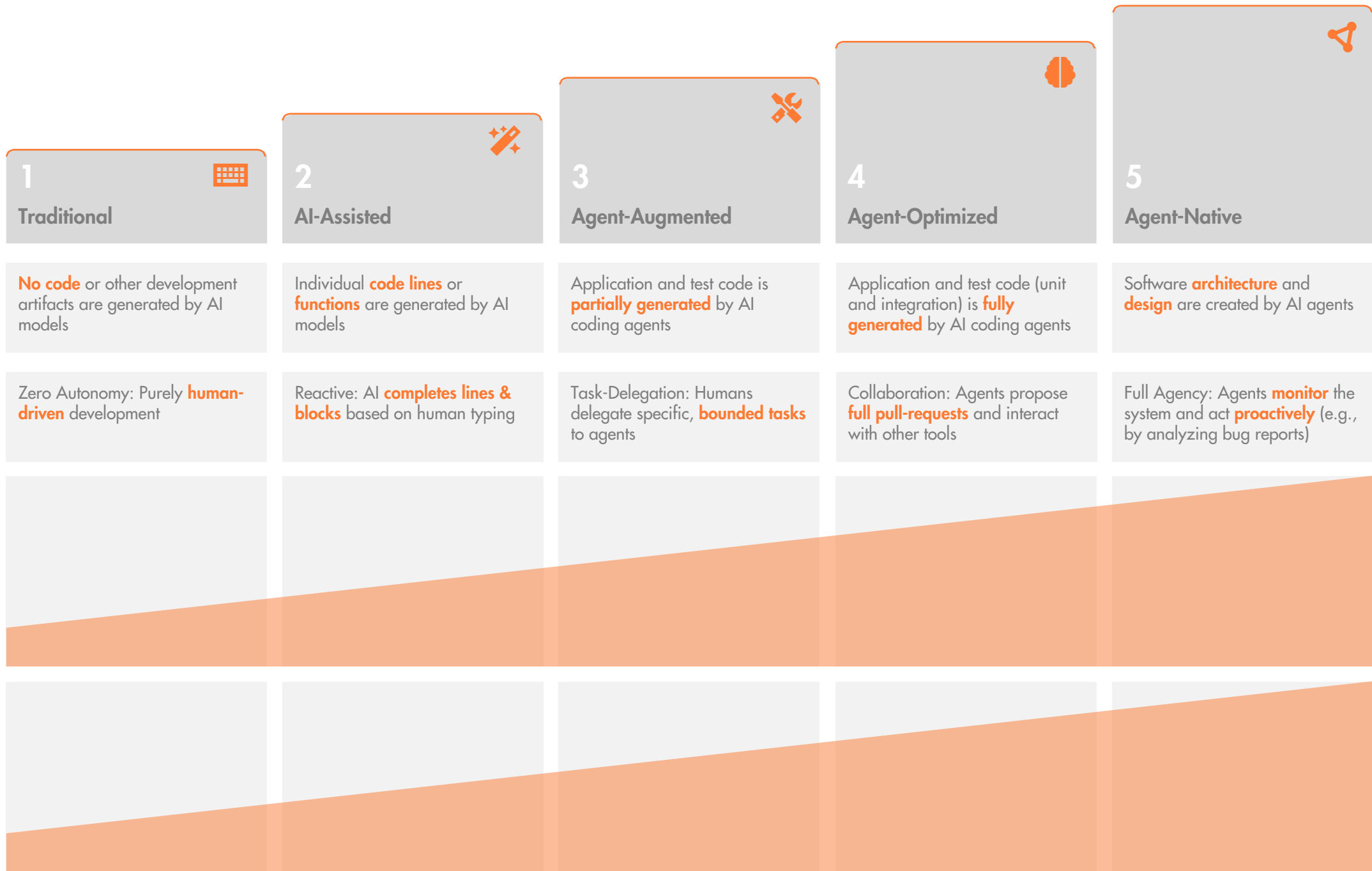
Ausblick & Takeaways



Scope
The engineering phases and artifacts handled by AI



Autonomy & Agency
The AI's degree of independence



 **Scope**
The engineering phases and artifacts handled by AI

 **Autonomy & Agency**
The AI's degree of independence

 **Quality Assurance**
The mechanisms safeguarding software quality

 **Governance**
The framework for compliance, financial oversight, traceability, and accountability

	<div>1</div> <div>Traditional</div> <div></div>	<div>2</div> <div>AI-Assisted</div> <div></div>	<div>3</div> <div>Agent-Augmented</div> <div></div>	<div>4</div> <div>Agent-Optimized</div> <div></div>	<div>5</div> <div>Agent-Native</div> <div></div>
<div> Scope</div> <div>The engineering phases and artifacts handled by AI</div>	No code or other development artifacts are generated by AI models	Individual code lines or functions are generated by AI models	Application and test code is partially generated by AI coding agents	Application and test code (unit and integration) is fully generated by AI coding agents	Software architecture and design are created by AI agents
<div> Autonomy & Agency</div> <div>The AI's degree of independence</div>	Zero Autonomy: Purely human-driven development	Reactive: AI completes lines & blocks based on human typing	Task-Delegation: Humans delegate specific, bounded tasks to agents	Collaboration: Agents propose full pull-requests and interact with other tools	Full Agency: Agents monitor the system and act proactively (e.g., by analyzing bug reports)
<div> Quality Assurance</div> <div>The mechanisms safeguarding software quality</div>	Quality assurance is manual or automated	Quality assurance is manual or automated	<ul style="list-style-type: none"> Quality assurance is fully automated Quality assurance includes at least: static analysis, unit and integration tests, and SCA An agent harness is established 	<ul style="list-style-type: none"> The agent fixes defects automatically The agent closes test gaps automatically The agent harness is actively improved 	<ul style="list-style-type: none"> The process for improving the agent harness is automated Responsibility for quality assurance is organizationally anchored
<div> Governance</div> <div>The framework for compliance, financial oversight, traceability, and accountability</div>	<ul style="list-style-type: none"> AI is not used for software development Individuals (e.g., developers, reviewers, testers) are accountable for results 	<ul style="list-style-type: none"> Approved model providers are defined Contracts with model providers are in place Individuals (e.g., developers, reviewers, testers) are accountable for results 	<ul style="list-style-type: none"> Policies for AI usage are in place Cost limits are enforced Individuals (e.g., developers, reviewers, testers) are accountable for results 	<ul style="list-style-type: none"> Cost monitoring provides breakdown for teams, agents, and tasks Metrics for AI adoption are defined Reviewers are accountable for results 	<ul style="list-style-type: none"> Full traceability is provided for every agent-generated change Ultimate accountability remains with a human

Hype und Kritik

Reifegradmodell für Agentic Software Engineering

▶ **Wo Unternehmen heute wirklich stehen**

Praxiserfahrungen aus unserer eigenen Entwicklung

Qualität, Governance und Verantwortlichkeit

Ausblick & Takeaways

1



Traditional

No **code** or other development artifacts are generated by AI models

Zero Autonomy: Purely **human-driven** development

2



AI-Assisted

Individual **code lines** or **functions** are generated by AI models

Reactive: AI **completes lines & blocks** based on human typing

3



Agent-Augmented

Application and test code is **partially generated** by AI coding agents

Task-Delegation: Humans delegate specific, **bounded tasks** to agents

4



Agent-Optimized

Application and test code is **fully generated** by AI coding agents

Collaboration: Agents propose **full pull-requests** and interact with other tools

5

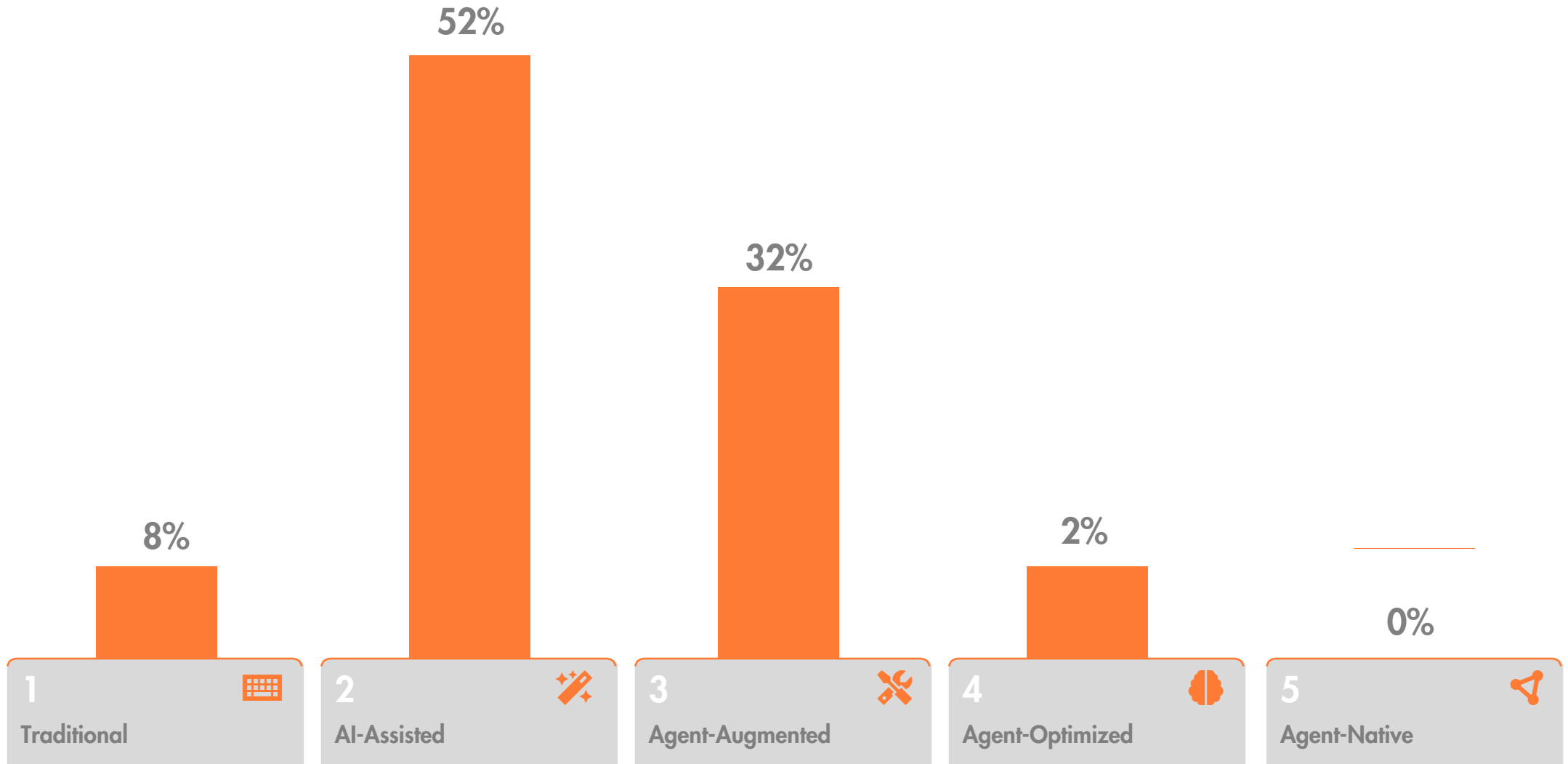


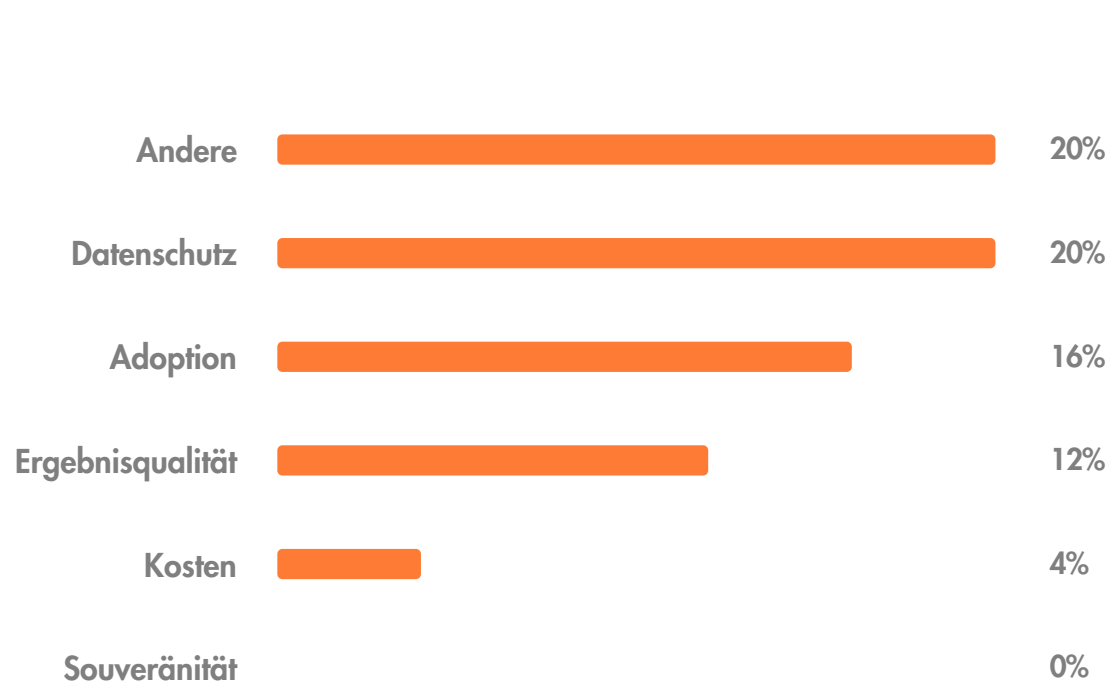
Agent-Native

Software **architecture** and **design** are created by AI agents

Full Agency: Agents **monitor** the system and act **proactively**

Wo steht ihr?





»Kein Agent-Tooling für ABAP«
»Langsame Unternehmensprozesse«
»Jedes Team macht was anderes, oder noch schlimmer, das gleiche unter anderem Namen.«
»Keine Zeit für Evaluierungen.«

»Nur On-Prem wegen VS-NfD«
»Codex zurückgerollt, Sandbox nicht sicher genug«

»Pilotphase ohne Erfolg beendet«
»Review-Agent findet immer noch etwas«

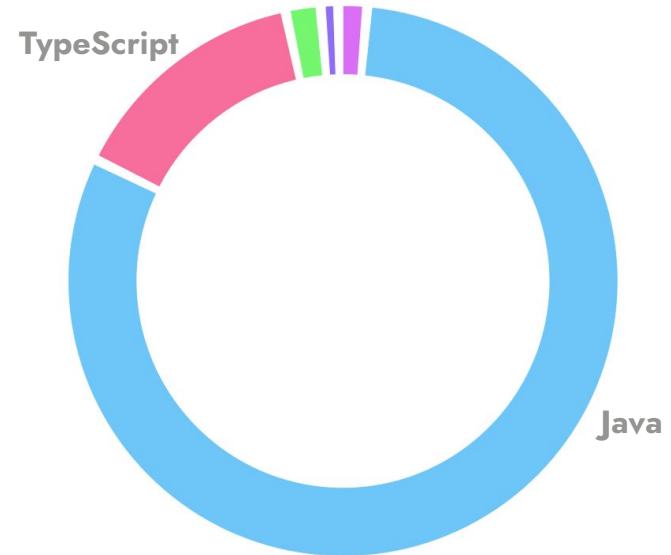


Praxiserfahrungen aus der eigenen
Softwareentwicklung

Was bauen wir?

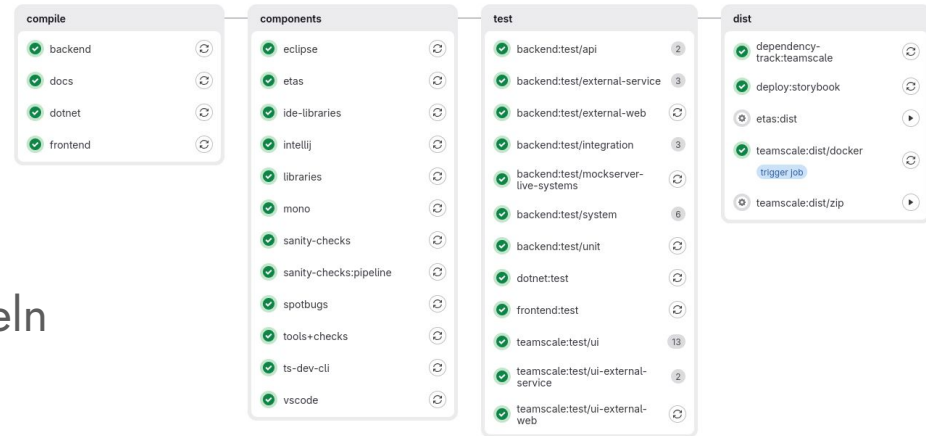
- Software Intelligence Platform
- ~2 Millionen Zeilen Code
- ~20 Köpfe Entwicklungsteam in 4 Squads
(plus Studierende, plus Support aus anderen Teams)
- SaaS und On-Prem

Qualität der Software hat für
uns sehr hohe Priorität



Qualitätssicherung und CI/CD (schon prä-AI)

- Feature-Branches
- CI/CD Pipeline mit 56 Jobs
 - Laufzeit 30-40 Minuten
 - Läuft für jeden Push
 - Bis zu 240 parallele VMs
- >33.000 automatisierte Tests
- Statische Analyse mit über 2.000 Regeln
- Überwachung von Testlücken
- Flächendeckende Merge-Request-Reviews



Timeline: Agentic Development

- Ende 2023/Anfang 2024
 - Erste Prototypen in Teamscale
 - Studien und Forschungsarbeiten

Timeline: Agentic Development

- Ende 2023/Anfang 2024
 - Erste Prototypen in Teamscale
 - Studien und Forschungsarbeiten
- Mitte 2025
 - Evaluierung und Freigabe für Dev-Team
 - Kaum ernsthafte Nutzung

Timeline: Agentic Development

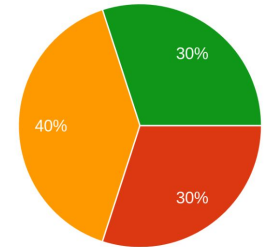
- Ende 2023/Anfang 2024
 - Erste Prototypen in Teamscale
 - Studien und Forschungsarbeiten
- Mitte 2025
 - Evaluierung und Freigabe für Dev-Team
 - Kaum ernsthafte Nutzung
- Januar 2026
 - Strukturierte Einführung von Claude Code
 - Einführungstraining mit externem Coach
 - 1 Woche “frei” für Hands-On-Erfahrungen
 - Nutzung in den Wochen danach gewünscht, selbst wenn es sich langsamer anfühlt

Timeline: Agentic Development

- Ende 2023/Anfang 2024
 - Erste Prototypen in Teamscale
 - Studien und Forschungsarbeiten
- Mitte 2025
 - Evaluierung und Freigabe für Dev-Team
 - Kaum ernsthafte Nutzung
- Januar 2026
 - Strukturierte Einführung von Claude Code
 - Einführungstraining mit externem Coach
 - 1 Woche "frei" für Hands-On-Erfahrungen
 - Nutzung in den Wochen danach gewünscht, selbst wenn es sich langsamer anfühlt

Ende Februar:

How often have you used Claude Code in your daily work over the last two weeks?



- Never
- Occasionally (once or twice a week)
- Regularly (daily for specific tasks)
- Heavily (it is my primary way of writing/refactoring code)

Agentic Development & Qualitätssicherung

Agentic Development & Qualitätssicherung

- Entwickler ist Verantwortlich für die Eingabequalität ins Review

Agentic Development & Qualitätssicherung

- Entwickler ist Verantwortlich für die Eingabequalität ins Review
- Tests dürfen/sollen durch KI geschrieben werden, werden aber inhaltlich geprüft

Agentic Development & Qualitätssicherung

- Entwickler ist Verantwortlich für die Eingabequalität ins Review
- Tests dürfen/sollen durch KI geschrieben werden, werden aber inhaltlich geprüft
- Agentische Reviews als ergänzende Maßnahme

Agentic Development & Qualitätssicherung

- Entwickler ist Verantwortlich für die Eingabequalität ins Review
- Tests dürfen/sollen durch KI geschrieben werden, werden aber inhaltlich geprüft
- Agentische Reviews als ergänzende Maßnahme
- Keine Abstriche bei Tests und Merge-Request-Reviews
 - Relativ gesehen entfällt jetzt viel mehr Zeit auf Reviews
 - Aber: Reviews finden jetzt z.T. zielgerichteter statt

Agentic Development & Qualitätssicherung

- Entwickler ist Verantwortlich für die Eingabequalität ins Review
- Tests dürfen/sollen durch KI geschrieben werden, werden aber inhaltlich geprüft
- Agentische Reviews als ergänzende Maßnahme
- Keine Abstriche bei Tests und Merge-Request-Reviews
 - Relativ gesehen entfällt jetzt viel mehr Zeit auf Reviews
 - Aber: Reviews finden jetzt z.T. zielgerichteter statt
- Deterministische Qualitätsprüfung als Teil des Harness
 - Statische Analyse
 - "Tripwires"

“Menschliche” Code Reviews

Review durch Experten, flexibel, komplexe Zusammenhänge

Agentische Reviews

Skalierbarer als Menschen, kann mehr Zusammenhänge berücksichtigen

Statische Analyse

Automatisch, deterministisch, preiswert, skaliert gut

Wo stehen wir?



Scope

The engineering phases and artifacts handled by AI



Autonomy & Agency

The AI's degree of independence



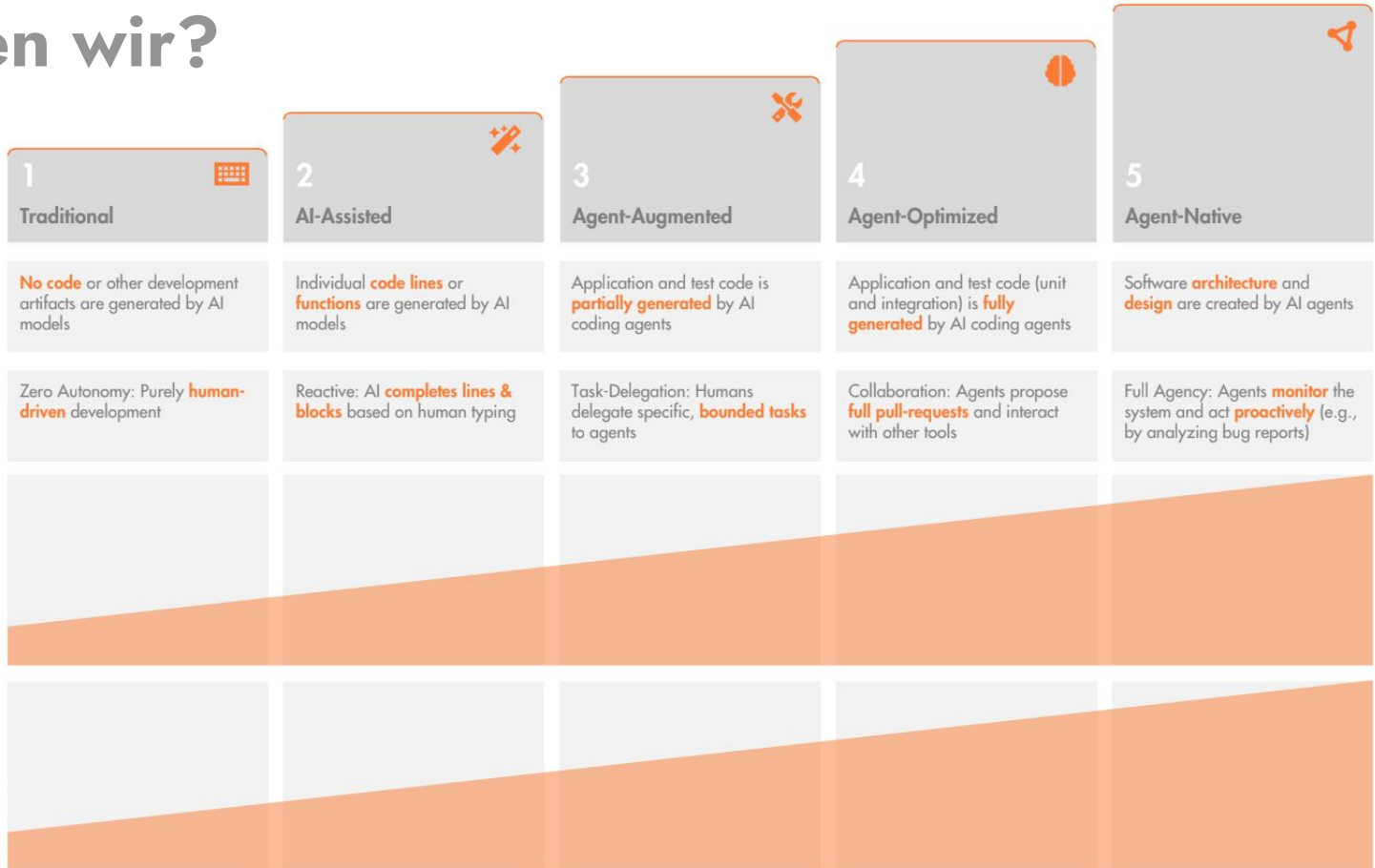
Quality Assurance

The mechanisms safeguarding software quality



Governance

The framework for compliance, financial oversight, traceability, and accountability



Wo stehen wir?



Scope

The engineering phases and artifacts handled by AI



Autonomy & Agency

The AI's degree of independence



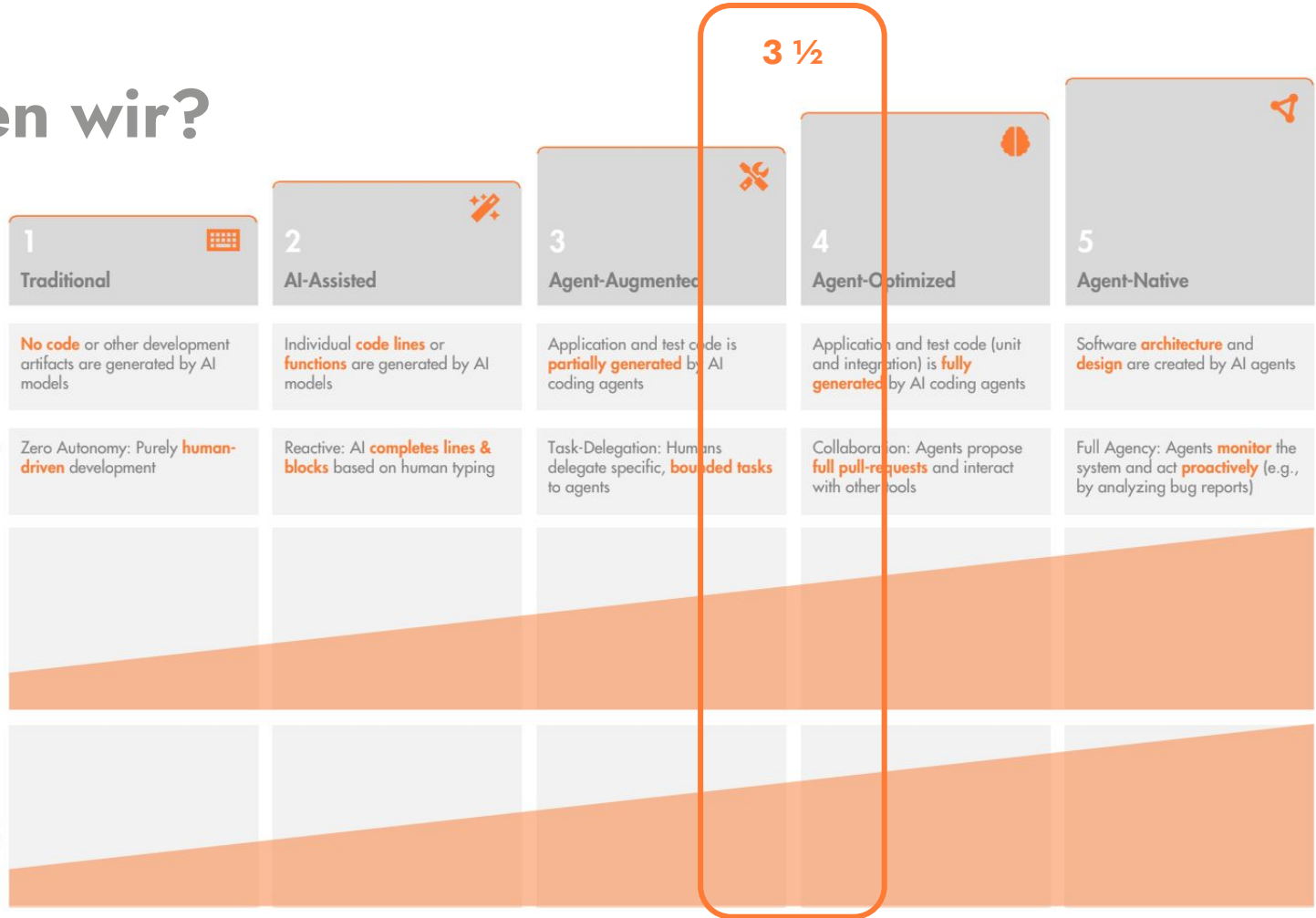
Quality Assurance

The mechanisms safeguarding software quality

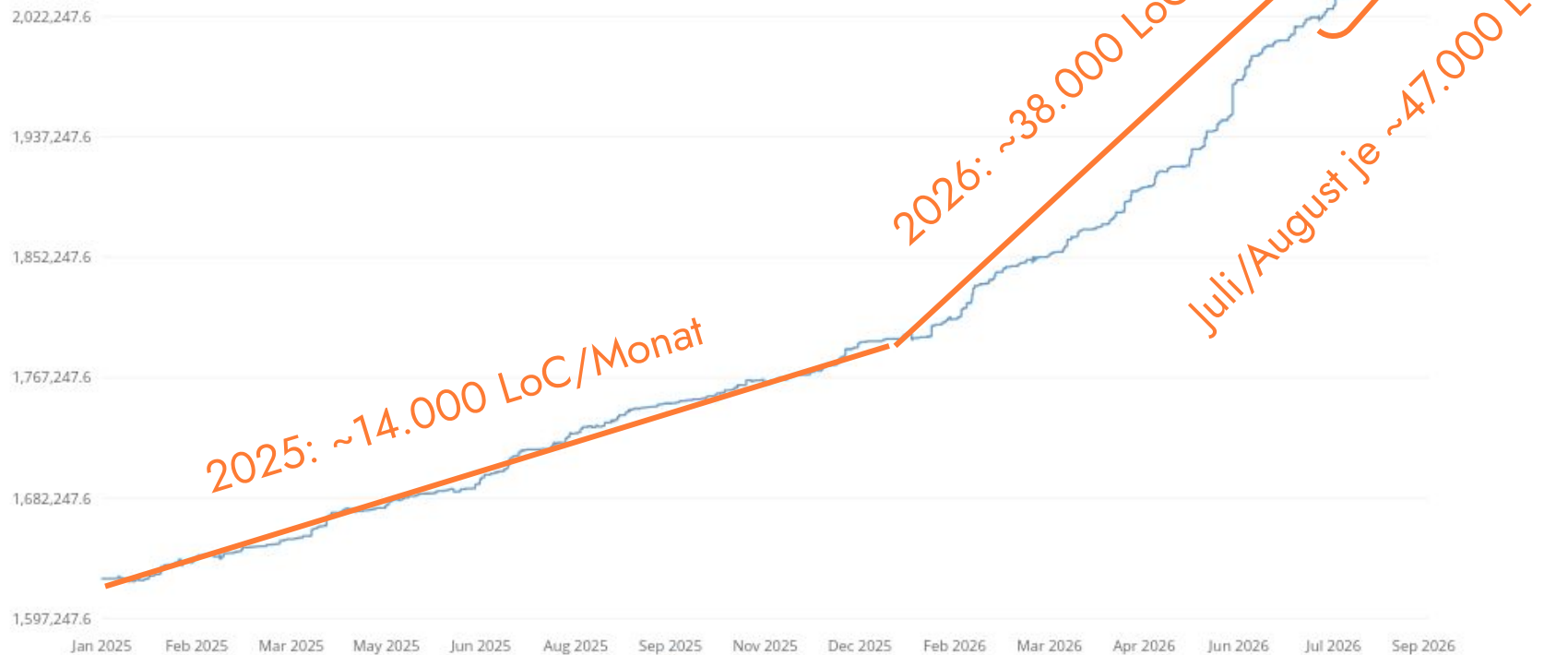


Governance

The framework for compliance, financial oversight, traceability, and accountability



Lines of Code



Macht KI unsere Entwicklung “besser”?

- Schwer zu quantifizieren
- Manche Zahlen sind einfach zu messen, haben aber nur beschränkte Aussagekraft

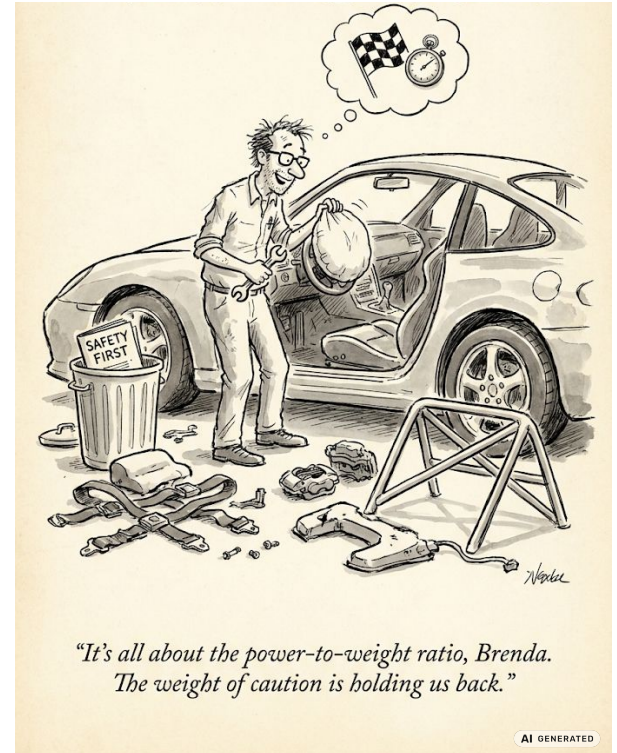
Macht KI unsere Entwicklung “besser”?

- Schwer zu quantifizieren
- Manche Zahlen sind einfach zu messen, haben aber nur beschränkte Aussagekraft
- Beispiel: Ticketdurchsatz um ca. 20% gestiegen, aber
 - Wir priorisieren und schneiden Tickets heute teilweise anders (wegen KI)
 - Tickets führen heute oft zu mehr (sinnvollen) Ergänzungen im Code

Bauchgefühl reicht von 30% bis 100% Effizienzgewinn

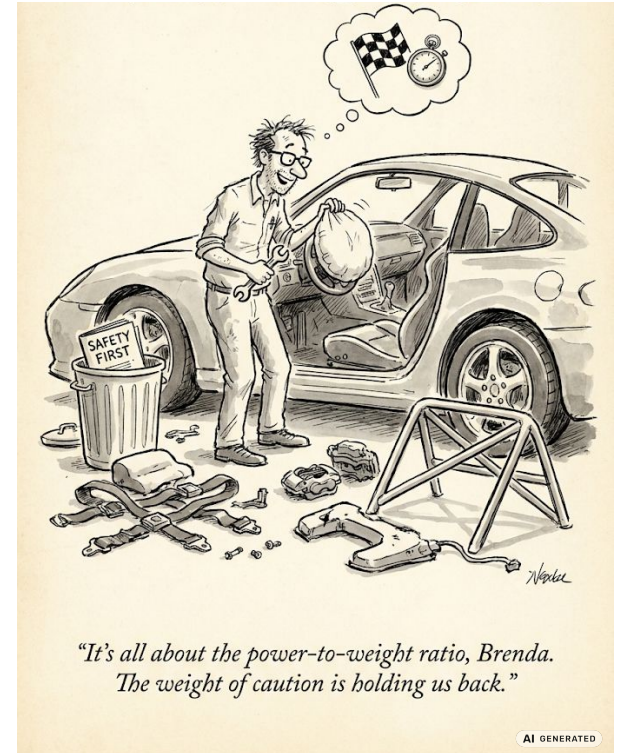
Ist Qualitätssicherung ein Bottleneck?

- Mehr Code und damit mehr QA sind eine Herausforderung
 - "Review Overload"
 - "Cognitive Surrender"



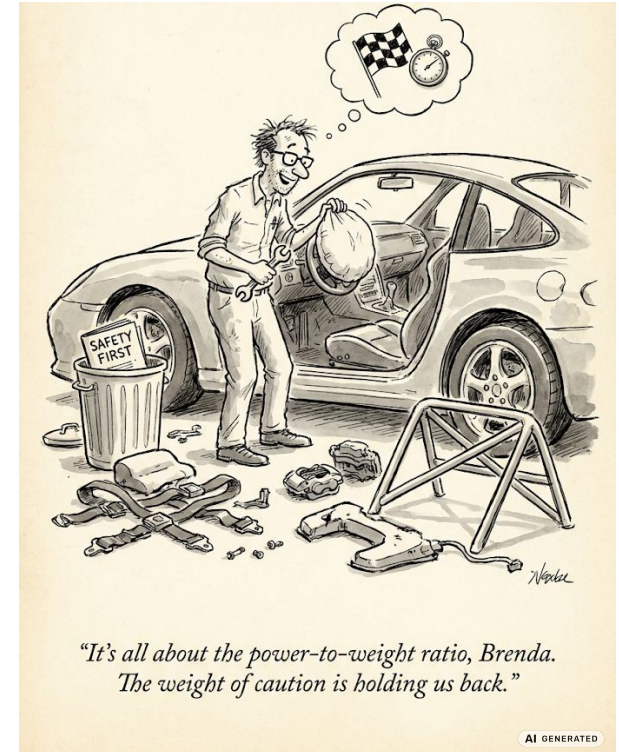
Ist Qualitätssicherung ein Bottleneck?

- Mehr Code und damit mehr QA sind eine Herausforderung
 - "Review Overload"
 - "Cognitive Surrender"
- Unser Eindruck ist aber nicht, dass wir etwas wegnehmen wollen, sondern eher dass wir mehr QA brauchen



Ist Qualitätssicherung ein Bottleneck?

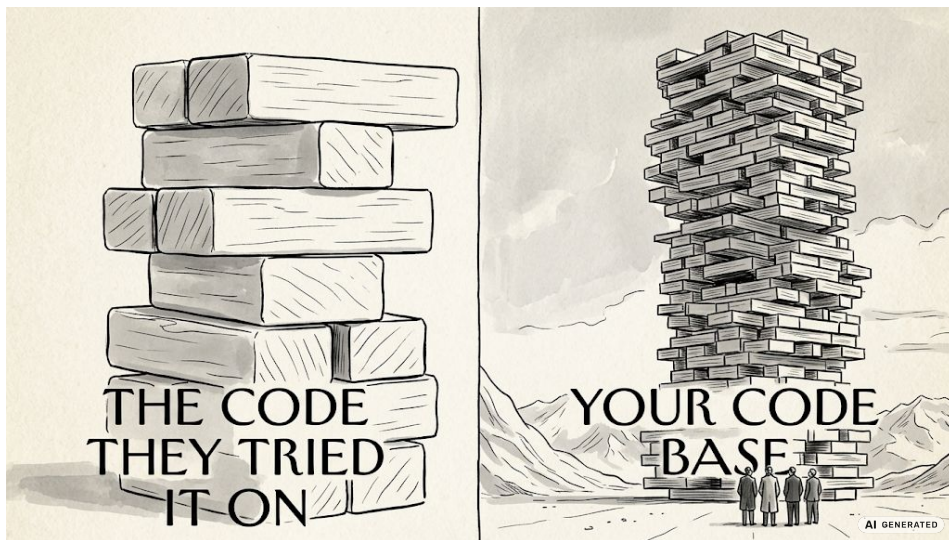
- Mehr Code und damit mehr QA sind eine Herausforderung
 - "Review Overload"
 - "Cognitive Surrender"
- Unser Eindruck ist aber nicht, dass wir etwas wegnehmen wollen, sondern eher dass wir mehr QA brauchen
- Aber auch klarer Shift zu noch mehr Automatisierung
 - Weniger Detailtiefe bei "menschlicher" QA
 - Mehr zusätzliche deterministische Guardrails



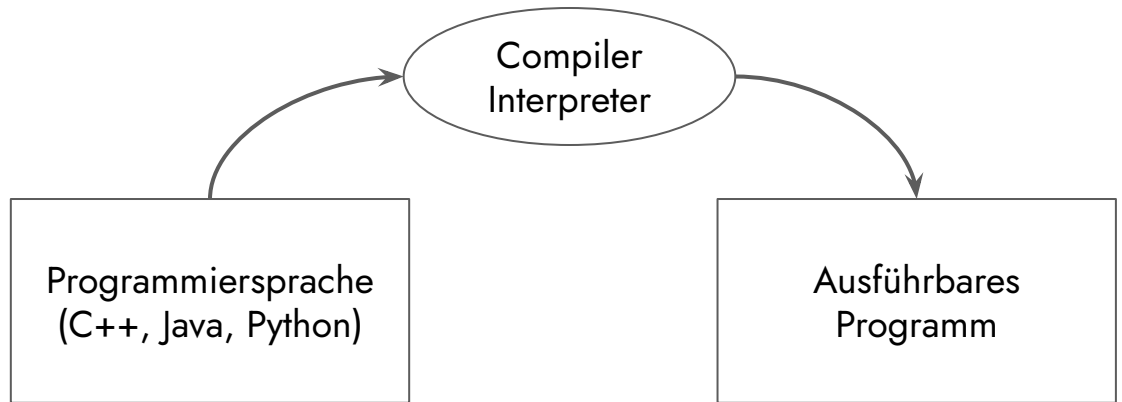
Takeaways

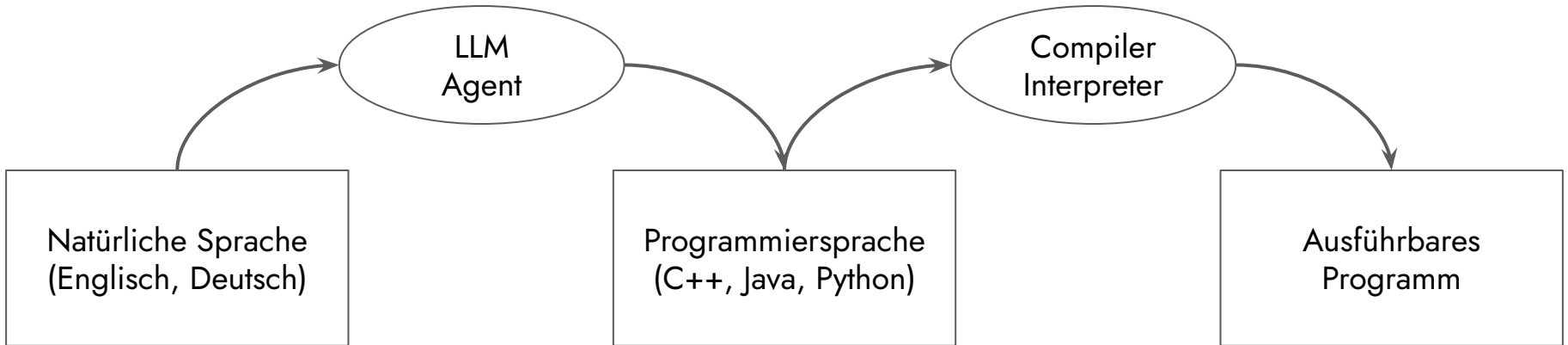
- KI-Einführung und Prozessänderungen brauchen einen **klaren Rahmen**
- KI funktioniert viel besser als **Ergänzung**, nicht als Ersatz von Menschen
- KI führt zwangsläufig zu **Verschiebungen bei der Qualitätssicherung**, sollte aber nicht zu Abstrichen führen

Qualität, Governance und Verantwortlichkeit



AI "Influencer" 2026

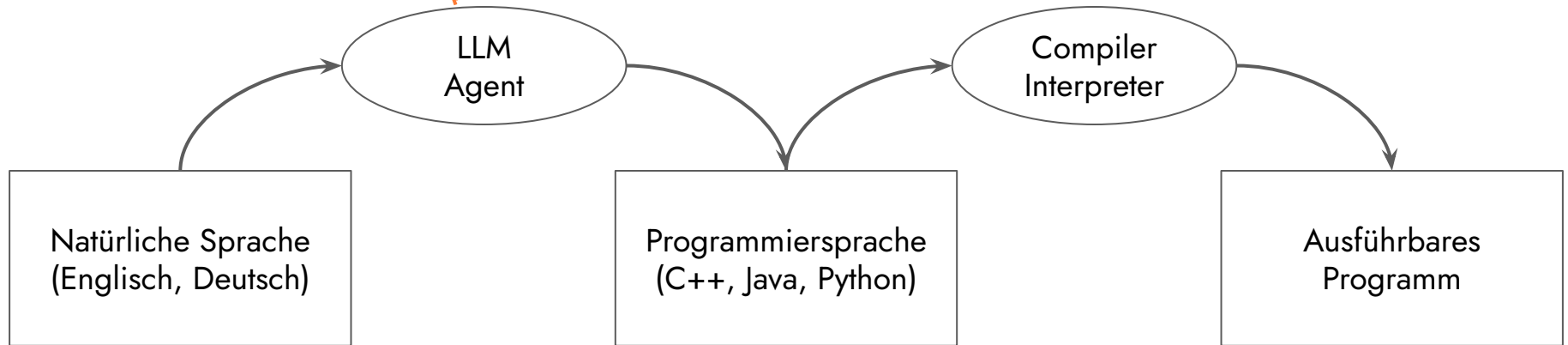




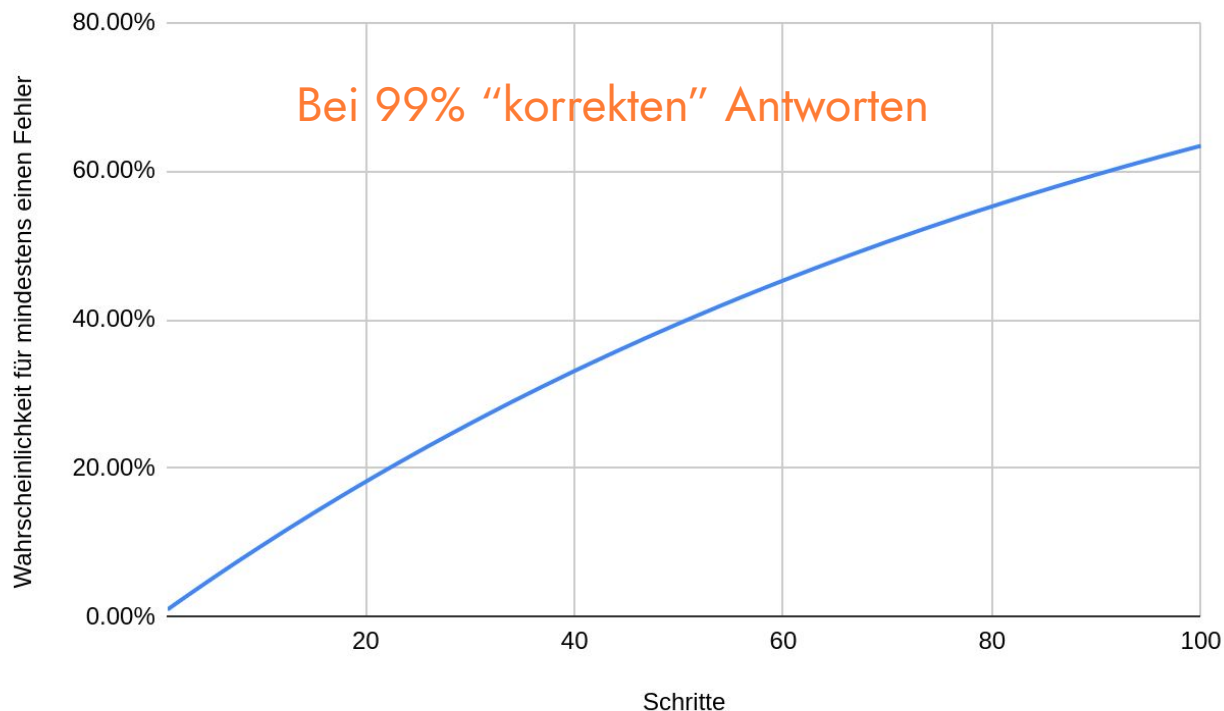
Reliance on Outputs and Actions. Artificial intelligence and large language models are frontier technologies that are still improving in accuracy, reliability and safety. When you use our Services, you acknowledge and agree:

1. Outputs may not always be accurate and may contain material inaccuracies even if they appear accurate because of their level of detail or specificity.
2. Actions may not be error free or operate as you intended.
3. You should not rely on any Outputs or Actions without independently confirming their accuracy.
4. The Services and any Outputs may not reflect correct, current, or complete information.
5. Outputs may contain content that is inconsistent with Anthropic's views.

<https://www.anthropic.com/legal/consumer-terms>

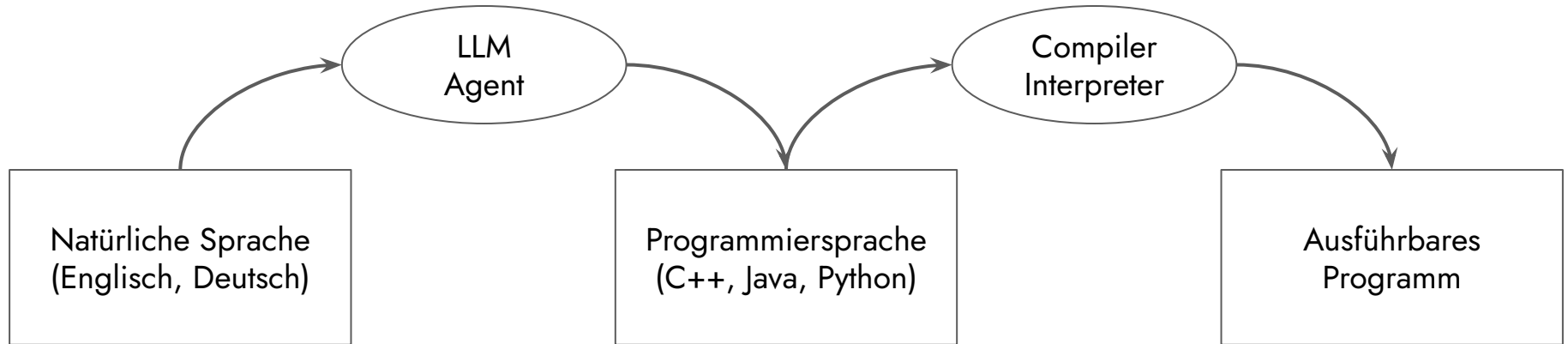


Fehlerwahrscheinlichkeiten für Multi-Step-Agenten



- Probabilistischer Prozess
- Neue Modellvarianten alle paar Monate
- Benchmarks mit geringer Aussagekraft
- Fehler sind Teil des Vertrags mit dem LLM-Anbieter

- Deterministisch
- Definierte Sprachstandards und Kompatibilität
- Umfassende Testsuites und Zertifizierungen für Compiler
- Compiler-Anbieter haftet für bestimmte Fehler



Zentrale Stellschraube: Risikobewertung

Zentrale Stellschraube: **Risikobewertung**

- Vertraulichkeit der Eingabedaten
 - Reicht der vertragliche Schutz im Vergleich zum Stellenwert des Codes?
 - Muss Datenschutz berücksichtigt werden?

Zentrale Stellschraube: Risikobewertung

- Vertraulichkeit der Eingabedaten
 - Reicht der vertragliche Schutz im Vergleich zum Stellenwert des Codes?
 - Muss Datenschutz berücksichtigt werden?
- Ausgabequalität
 - Wie schlimm wirken sich Fehler im System aus?
 - Wie schlimm wäre es, wenn die Software an einem bestimmten Punkt nicht mehr funktioniert oder nicht mehr erweiterbar ist?

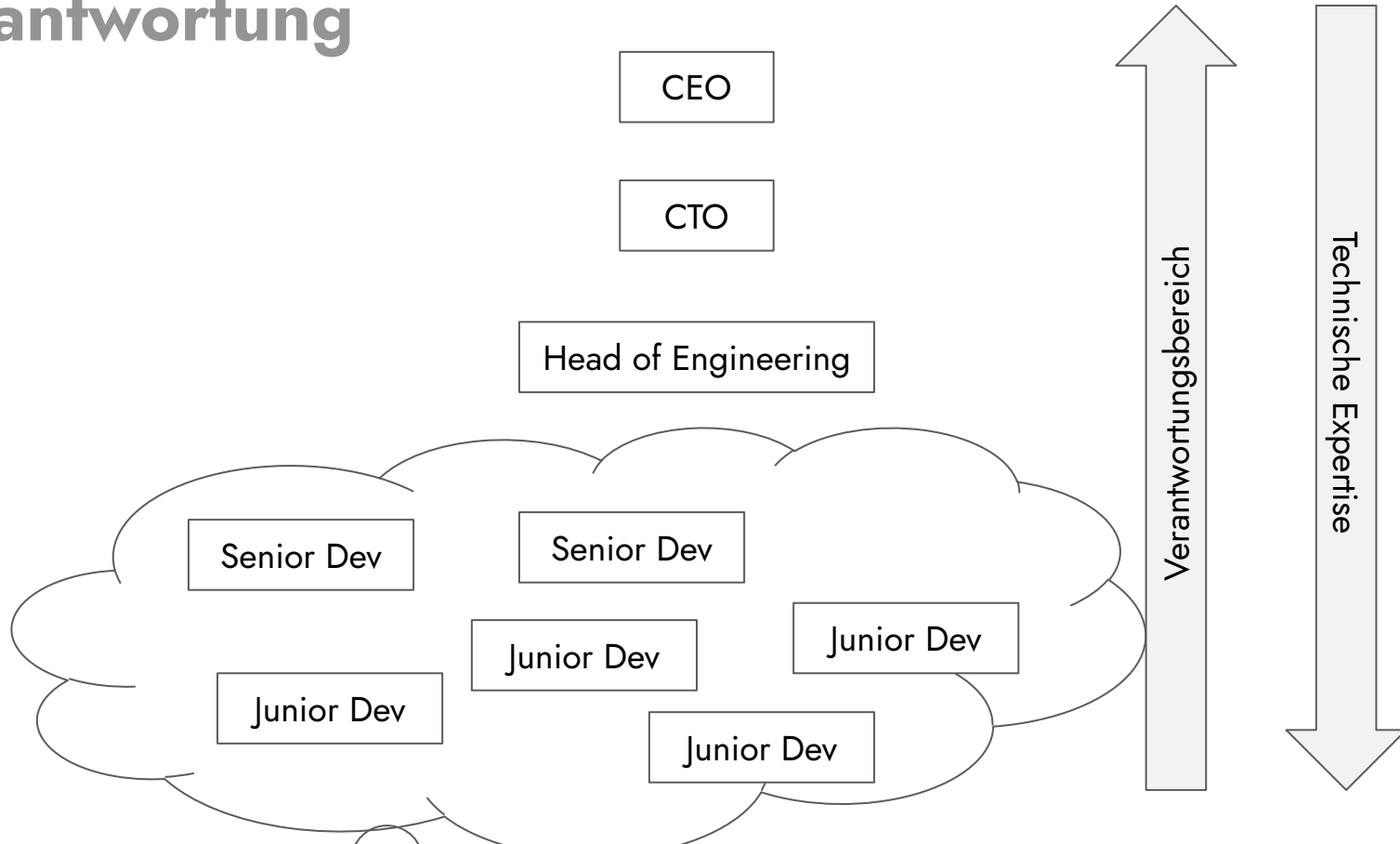
Zentrale Stellschraube: Risikobewertung

- Vertraulichkeit der Eingabedaten
 - Reicht der vertragliche Schutz im Vergleich zum Stellenwert des Codes?
 - Muss Datenschutz berücksichtigt werden?
- Ausgabequalität
 - Wie schlimm wirken sich Fehler im System aus?
 - Wie schlimm wäre es, wenn die Software an einem bestimmten Punkt nicht mehr funktioniert oder nicht mehr erweiterbar ist?
- Abhängigkeit in der Entwicklung
 - Was passiert, wenn die LLMs/Agenten länger nicht zur Verfügung stehen?
 - Ist eine Fehlerbehebung auch ohne KI noch möglich?

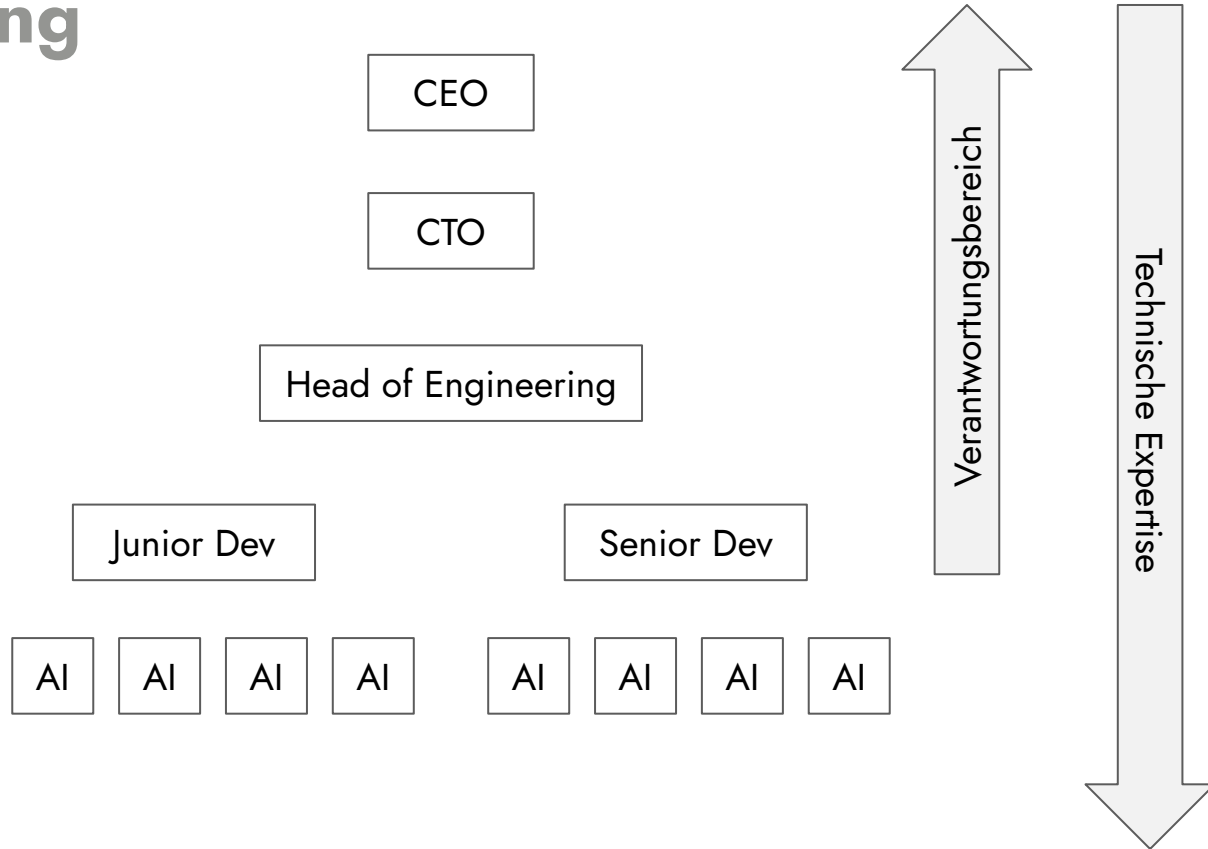
Zentrale Stellschraube: Risikobewertung

- Vertraulichkeit der Eingabedaten
 - Reicht der vertragliche Schutz im Vergleich zum Stellenwert des Codes?
 - Muss Datenschutz berücksichtigt werden?
- Ausgabequalität
 - Wie schlimm wirken sich Fehler im System aus?
 - Wie schlimm wäre es, wenn die Software an einem bestimmten Punkt nicht mehr funktioniert oder nicht mehr erweiterbar ist?
- Abhängigkeit in der Entwicklung
 - Was passiert, wenn die LLMs/Agenten länger nicht zur Verfügung stehen?
 - Ist eine Fehlerbehebung auch ohne KI noch möglich?
- Anforderungen an die Verfügbarkeit
 - Wie schnell muss auf Probleme im Betrieb reagiert werden?
 - In welchem Umfang sind Datenverluste akzeptabel?

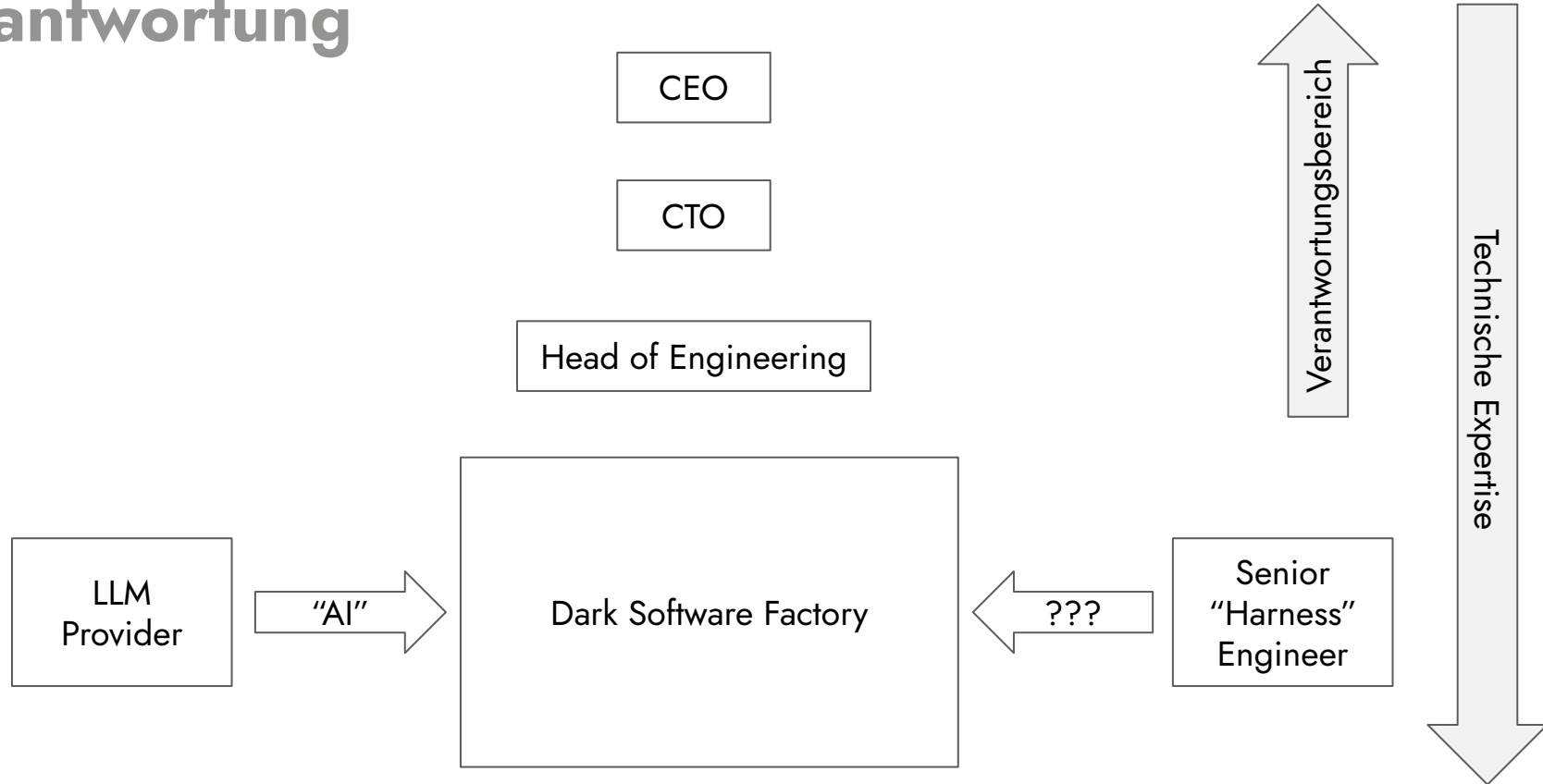
Verantwortung



Verantwortung



Verantwortung



Verantwortung auf Entwicklerebene



Tiefes Verständnis des Codes
(v.a. Reviews)

Indirekte Qualitätsindikatoren
(Tests, statische Analyse)

Ausschnitt
des Codes

Verantwortung auf Entwicklerebene



Partielles Verständnis des Codes
(gezielte Reviews)

Indirekte Qualitätsindikatoren
(Tests, statische Analyse)

Ausschnitt des Codes

Takeaways

- KI ist **kein Compiler** von “Sprache” zu Code
- Prozesse und damit auch Governance und Verantwortung müssen **grundsätzlich neu gedacht** werden
- Verantwortung wird “breiter und flacher”,
dadurch werden **indirekte Qualitätsindikatoren** immer wichtiger

Hype und Kritik

Reifegradmodell für Agentic Software Engineering

Wo Unternehmen heute wirklich stehen

Praxiserfahrungen aus unserer eigenen Entwicklung

Qualität, Governance und Verantwortlichkeit

▶ **Ausblick & Takeaways**



1. **»No Silver Bullet«:** Wir unterschätzen die inhärente Komplexität der Software-Entwicklung
2. **Jevons-Paradox:** Fähigkeitsgewinne werden für die Entwicklung noch komplexerer Systeme genutzt

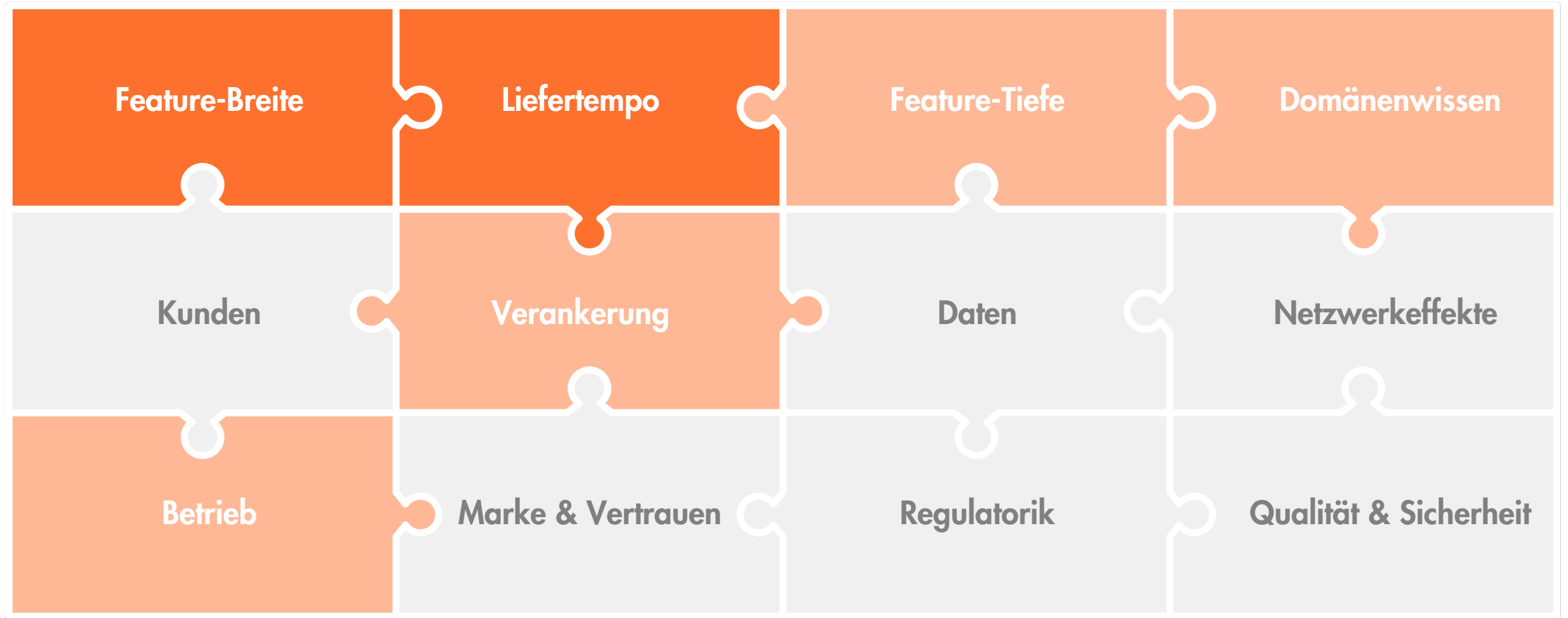


The code was never ^{only} the moat.

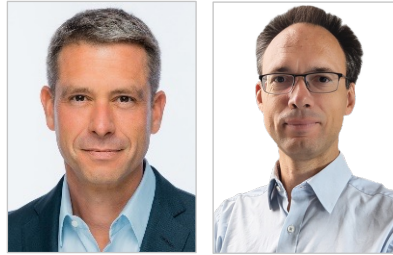


 Betroffen  Teilweise betroffen  Nicht betroffen

only
The code was never the moat.



Betroffen Teilweise betroffen Nicht betroffen



Die **Verantwortung**
bleibt beim **Menschen**.



Die LinkedIn-Bubble ist
nicht die **Realität**.

Mehr **Autonomie**
erfordert mehr **Qualität**
und **Governance**.

Der **Code** war nie der
(alleinige) »**Moat**«.



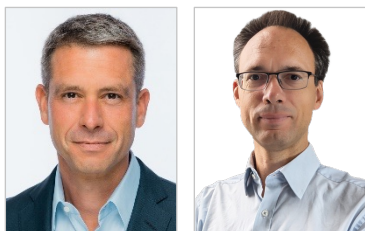
1
Traditional

2
AI-Assisted

3
Agent-Augmented

4
Agent-Optimized

5
Agent-Native



Dr. Florian Deissenböck · deissenboeck@cqse.eu · +49 179 7857188

Dr. Benjamin Hummel · hummel@cqse.eu · +49 176 10150006

CQSE GmbH
Centa-Hafenbrädl-Straße 59
81249 München

CQSE
Continuous Quality in Software Engineering

Softwarequalität trotz, wegen und mit KI

Wie agentische Entwicklung und verlässliche Softwarequalität zusammenfinden

Mittwoch

11. November

10:30 bis 12:00 Uhr

online und kostenlos – mit Aufzeichnung

Jetzt anmelden: tmscl.me/sit2611-2610



Dr. Benjamin Hummel