

KI in der Software-Entwicklung:

Unser Starter-Guide für Führungskräfte

slashwhy

Wo ist KI in der Entwicklung sinnvoll?

Warum sollte die KI nicht alles alleine machen?

Und welche Regeln wünscht sich dein Team?

Agenda & Vorwort

Viele Unternehmen stehen aktuell vor derselben Herausforderung: KI-Tools halten Einzug in die Entwicklungsteams. Doch es fehlt an klaren Strukturen, Verantwortlichkeiten und Entscheidungsrahmen. Zwischen Experimentierfreude und Unsicherheit, zwischen Effizienzgewinn und Qualitätsanspruch entsteht ein Spannungsfeld, das Orientierung braucht.

Mit diesem Paper wollen wir genau an dieser Stelle unterstützen und liefern konkrete Impulse für Führungskräfte, die den KI-Wandel nicht nur zulassen, sondern aktiv gestalten möchten.

Kapitel 1

03 Was ist AI-Assisted Development?

Kapitel 2

05 Status Quo aus Sicht der Entwickler:innen

07 Status Quo aus Sicht der Unternehmen

08 Status Quo aus Sicht der Führungskräfte

Kapitel 3

10 Warum die KI nicht die ganze Arbeit machen sollte

13 Wo KI in der Software Entwicklung sinnvoll ist

Kapitel 4

15 Hilfreiche GenAI-Tools für Software-Entwickler:innen

Kapitel 5

13 Tipps für Führungskräfte „AI-Assisted Development“

21 Fazit

Kapitel 1 / Einstieg

slashwhy

Was ist AI-Assisted Development?

Besonders in unternehmensinternen Software-Projekten zählt heute vor allem eines: schnell erste funktionierende Ergebnisse zu liefern. Bei neuen Ideen, in Prototypen oder frühen Projektphasen müssen Teams zeigen, dass ihre Ansätze tragfähig sind, bevor es in die Ausarbeitung geht.

Bisher bedeutete das für Entwickler:innen viel Grundlagenarbeit: recherchieren, das Projekt-Setup aufsetzen, Standardcode schreiben oder erste Funktionen testweise implementieren. Diese Aufgaben sind wichtig, binden aber Zeit, die bei Themen wie Software-Architektur, Code-Qualität und Sicherheit oft besser investiert wäre. **Genau hier setzt AI-Assisted Development an. Gemeint ist der kontrollierte Einsatz von KI-Tools in der Software-Entwicklung.**

Statt die Verantwortung an die KI abzugeben, wie es beim sogenannten Vibe Coding der Fall ist, werden Tools wie ChatGPT, Cursor, Claude, GitHub Copilot, Amazon Code-Whisperer oder Gemini deutlich gezielter genutzt. Sie helfen beim Generieren von Code-Vorschlägen oder bei der Beschleunigung repetitiver Aufgaben. Die Entscheidungs- und Gestaltungsverantwortung bleibt aber bei den Entwickler:innen. **KI wird bewusst als Werkzeug eingesetzt, nicht als alleiniger Lösungsgeber.**

„Professionelle Software-Entwicklung braucht mehr als lauffähigen Code. Sie braucht Struktur, Austausch, Qualitätssicherung und Teams, die wissen, was sie tun.“

Dr. Joachim Wilharm

Geschäftsführer slashwhy



Kapitel 2 / Status-Quo

slashwhy

GenAI-Nutzung der Entwickler:innen

- ✓ Code schreiben oder vervollständigen
- ✓ Bei Tests unterstützen
- ✓ Refactoring-Vorschläge und technische Dokumentation beschleunigen
- ✓ Erste Architekturvorschläge und Abfragen an APIs oder Datenbanken erstellen

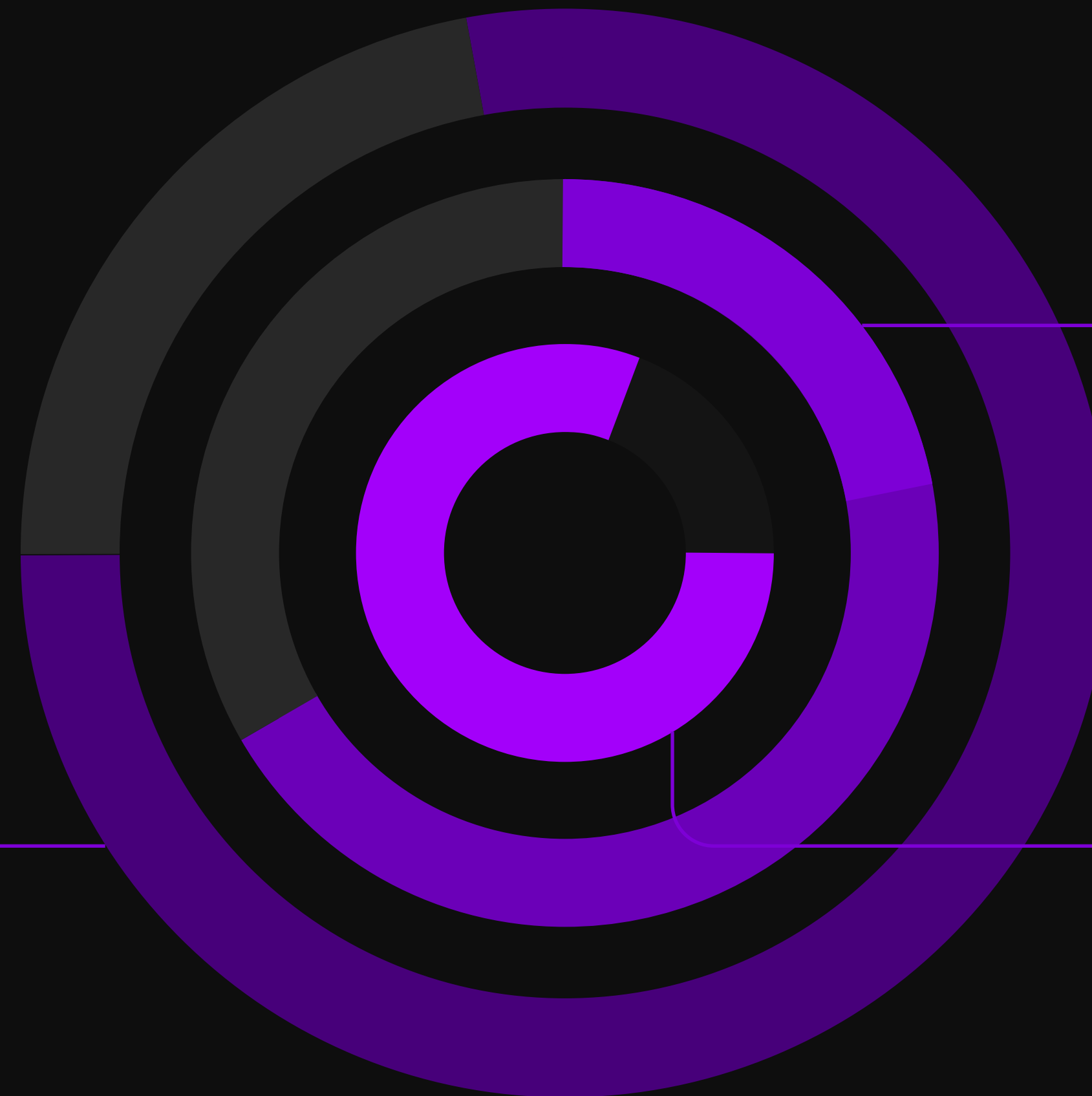
Künstliche Intelligenz hat sich in der Software-Entwicklung längst etabliert. Was noch vor wenigen Jahren als Zukunftsvision galt, ist heute für viele Entwickler:innen fester Bestandteil ihrer täglichen Arbeit. Die Stack Overflow Developer Survey 2024 untermauert diesen Eindruck mit konkreten Zahlen (siehe nächste Seite). Demnach nutzen oder planen 76% der Entwickler:innen den Einsatz von KI-gestützten Lösungen.

Auch eine GitHub-Umfrage aus dem Jahr 2024 bestätigt diesen Trend. Folgt man dieser von einer Software-Entwicklungsplattform in Auftrag gegebenen Studie, haben sogar 97 Prozent der befragten Entwickler:in-

nen KI-Coding-Tools bereits in ihrer Arbeit eingesetzt. In jedem Fall zeichnet sich ab, dass generative KI in der Entwicklungspraxis angekommen ist. Nicht nur als Spielerei, sondern als Werkzeug mit echtem Einfluss auf Prozesse, Geschwindigkeit und Ergebnisse.

Dabei verschiebt sich die Arbeitsweise vieler Entwickler:innen spürbar. Statt alles von Hand zu schreiben, kann Code im Dialog mit dem Tool entstehen. Der Fokus liegt weniger auf der Implementierung selbst, sondern stärker auf der Formulierung von Anforderungen, der Interpretation von Vorschlägen und der Entscheidung, was davon sinnvoll ist.

Einblick in die Stack Overflow Developer Survey



76 %

der Entwickler:innen nutzen oder planen den Einsatz von KI-gestützten Lösungen

62 %

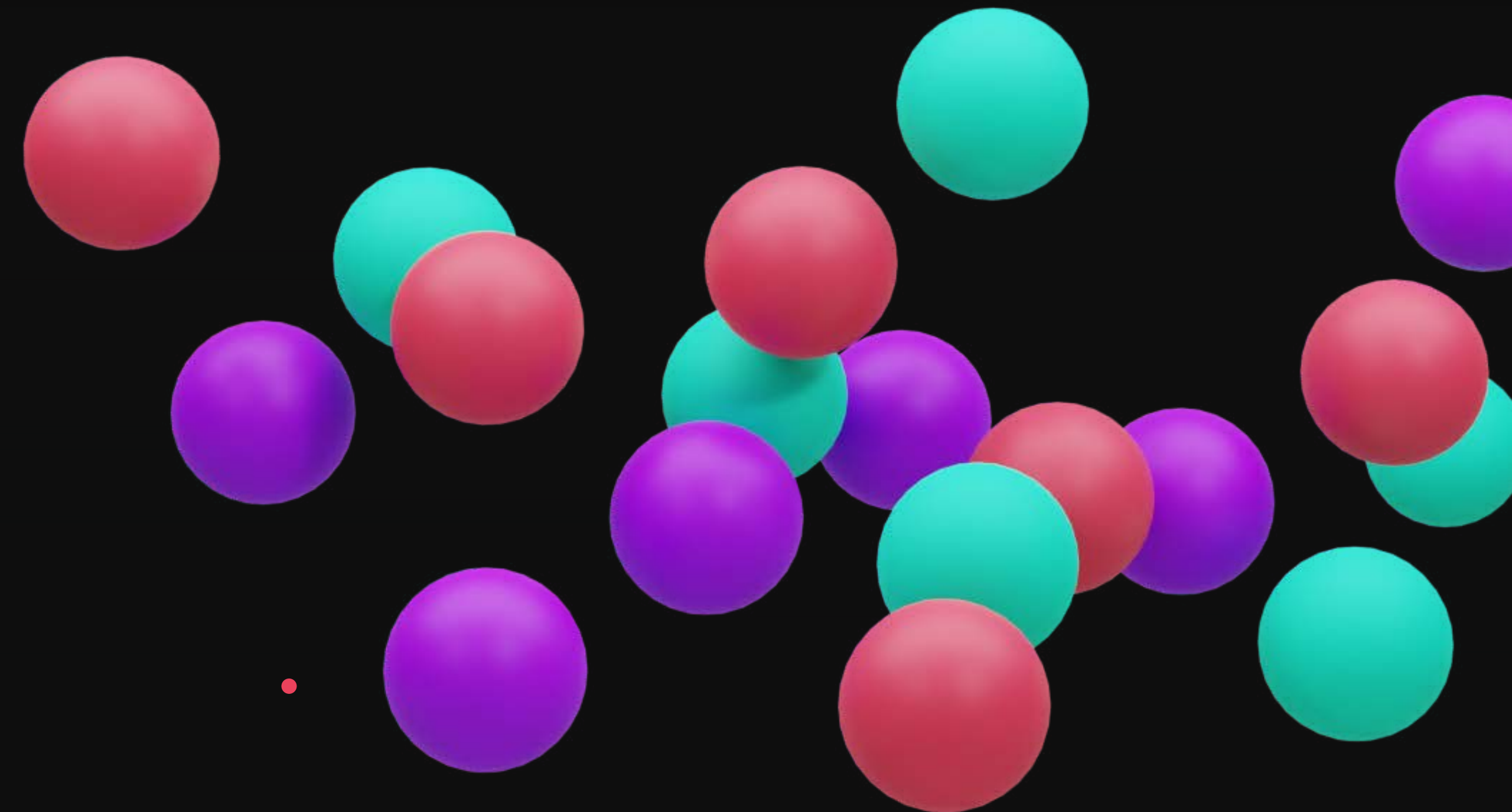
arbeiten bereits aktiv mit entsprechenden Tools; ein deutlicher Anstieg gegenüber 22 % im Jahr 2023

81 %

nennen Produktivitätssteigerung als größten Vorteil

Unsere Einschätzung

Viele Entwickler:innen stehen KI-Tools grundsätzlich offen gegenüber und wünschen sich mehr Orientierung sowie einen besseren organisatorischen Rahmen. Nicht, weil sie Regeln lieben, sondern weil sie Verantwortung für die neue Technologie übernehmen wollen, um möglichst effizient mit GenAI Tools arbeiten zu können und gleichzeitig ihre qualitativen (Team-) Standards aufrechtzuerhalten oder sogar zu erhöhen



Status-Quo der Unternehmen

Man muss nicht direkt über eine revolutionäre Veränderung des eigenen Geschäftsmodells nachdenken, um von KI zu profitieren. Es gibt unzählige GenAI-Tools, die das eigene Team bei alltäglichen Aufgaben entlasten können und produktiver machen.

Trotz der zunehmenden Nutzung von KI-Tools fehlt Unternehmen eine klare Strategie für deren Einsatz. Aktuelle Studien zeigen: Viele stecken noch in Pilotphasen, ohne dass klare Prozesse, Rollen oder Leitplanken etabliert sind.

Laut PwC AI Predictions 2024 sehen 73 Prozent der befragten Unternehmen KI als entscheidend für die eigene Wettbewerbsfähigkeit an.

Gleichzeitig geben nur 35 Prozent an, dass ihr Unternehmen aktuell über eine KI-Governance verfügt.

Der Mangel an organisatorischer Vorbereitung bremst dabei nicht nur den Wertbeitrag von KI, er birgt auch operative Risiken.

Die Accenture Technology Vision 2024 geht noch weiter: Dort wird betont, dass sich die Einführung von Generativer KI in den kommenden zwei Jahren in allen Unternehmensbereichen durchsetzen wird, aber nur Unternehmen mit klar definiertem Ordnungsrahmen („Responsible AI“) und funktionierenden Schnittstellen zwischen Business, IT und Produktentwicklung werden langfristig davon

profitieren. Gleichzeitig zeigt die Studie, dass generative KI das Potenzial hat, 44 Prozent aller Arbeitsstunden in den USA zu beeinflussen, Produktivitätssteigerungen in 900 unterschiedlichen Berufen zu ermöglichen und global ökonomische Wertschöpfung von mindestens 8 Billionen US-Dollar zu schaffen.

Branchen wie Finanzdienstleistung, Energie und Gesundheitswesen stehen vor dem großen Dilemma, die enormen Potenziale durch GenAI und ihre hohen Anforderungen an Sicherheit, Nachvollziehbarkeit und Governance in Einklang zu bringen. Der Mittelstand, besonders im verarbeitenden Gewerbe oder im

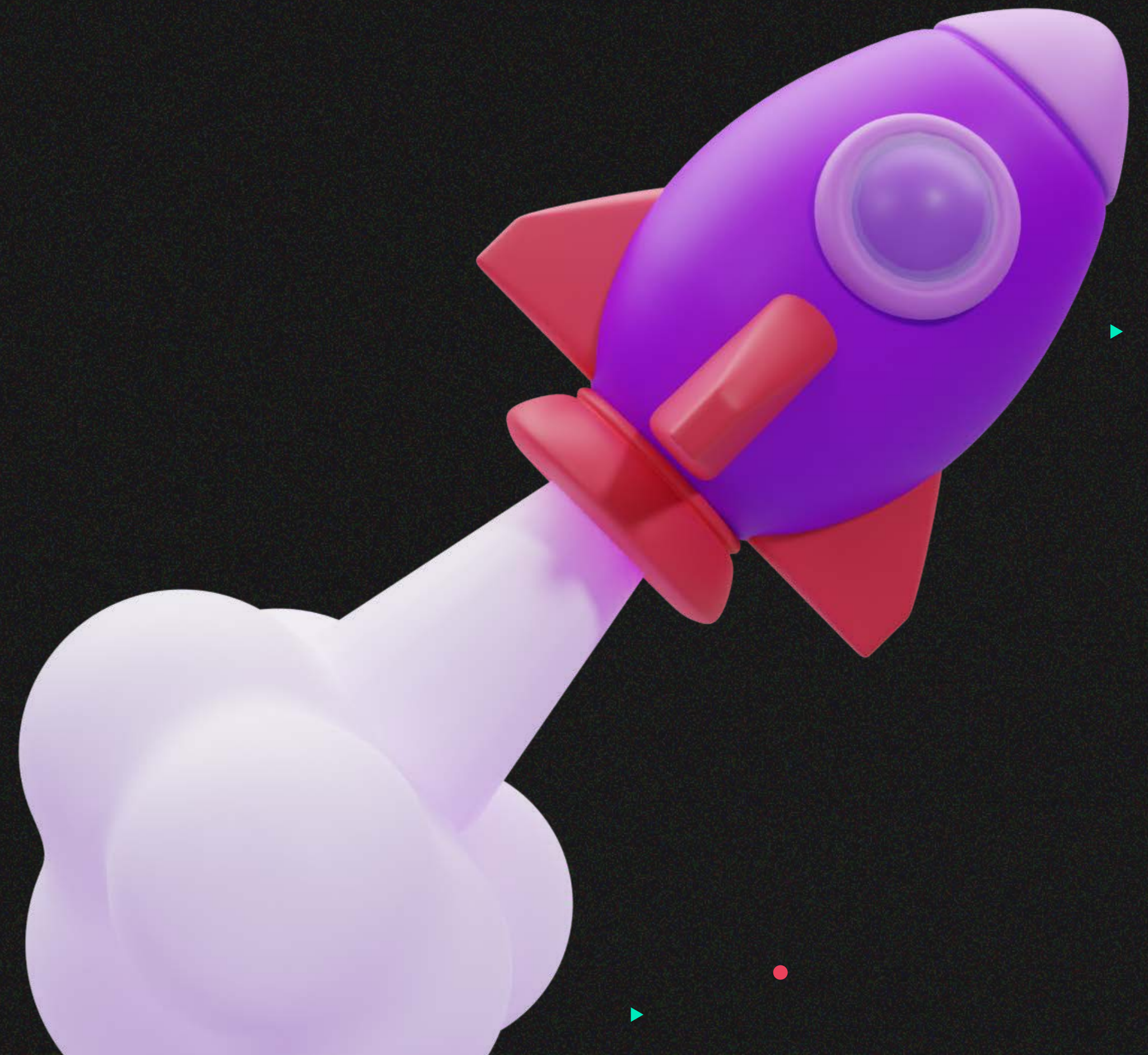
Projektgeschäft, experimentiert viel. Aber meist punktuell und weniger strategisch.

Vereint sind alle durch die zentrale Frage: Wo und wie fangen wir am besten an?

Unsere Einschätzung

Die Technologie ist bereit. Die Organisationen sind es oft noch nicht. Nicht aus Desinteresse, sondern weil Rahmenbedingungen und Zuständigkeiten fehlen. In Europa und besonders in Deutschland dominieren Vorsicht und Regulierungsdruck. Datenschutz, Urheberrecht und Compliance bremsen Innovation nicht grundsätzlich, verzögern sie aber oft unnötig. Vor allem dort, wo konkrete Anwendungsszenarien fehlen.

Im internationalen Vergleich sind die USA und China deutlich weiter, was Geschwindigkeit und Skalierung betrifft. Dort ist KI längst integraler Bestandteil digitaler Produkte und Service.



Sicht der Führungskräfte

So viel Potenzial AI-Assisted Development bietet, so nachvollziehbar sind auch die kritischen Stimmen, die sich in vielen Unternehmen regen. Gerade Teamleitungen, Senior Architekt:innen oder Head-of's äußern regelmäßig fundierte Bedenken. Sicher erkennst du die unten aufgeführten Fragen wieder. Wir sind überzeugt, dass diese kritischen Stimmen Gehör verdienen. Sie zeigen, dass es nicht reicht, neue Werkzeuge einzuführen, ohne sich über deren Auswirkungen auf Qualität, Kompetenz und Zusammenarbeit Gedanken zu machen.

Gleichzeitig sprechen sie für eine hohe Verantwortung, die Führungskräfte, Architekt:innen und Teams in der Gestaltung moderner Entwicklungsprozesse tragen.

Und: Die kritischen Stimmen berühren zentrale Aspekte moderner Software-Entwicklung wie das eigenständige Lösen komplexer Aufgaben, die verantwortungsvolle Gestaltung langlebiger Systeme und die nachhaltige Sicherung von Qualität und Verlässlichkeit im Code.

„Verlieren wir wichtige Kompetenzen, wenn sich unsere Entwickler:innen zu sehr auf KI verlassen?“

„Wer sorgt dafür, dass wir nicht schneller fehlerhaften Code produzieren?“

„Wie behalten wir Kontrolle über Qualität, Wartbarkeit und Sicherheit unserer Systeme?“

„Wofür bezahlen wir unsere Entwickler:innen, wenn die KI den Job macht?“

Unsere Einschätzung

Die kritischen Fragen, die viele Führungskräfte aktuell rund um AI-Assisted Development stellen, sind kein Zeichen von Skepsis, sondern von Verantwortung. Wer sie stellt, ist nicht der Bremser im System, sondern der Möglichmacher. Denn genau hier liegt der Unterschied zwischen blindem Tool-Einsatz und produktiver, verantwortungsvoller Einführung:

Folgen auf das Hinterfragen auch konkrete Leitplanken? Oder bleibt es bei Bauchgefühl und Einzelentscheidungen? Nur wer den Rahmen

definiert, kann auch entscheiden, wie viel Autonomie, wie viel Kontrolle und wie viel Veränderung wirklich sinnvoll ist. Jetzt ist der richtige Moment, aus kritischen Fragen klare Spielregeln zu machen.





KI kann hilfreich im
Entwicklungsprozess
sein. Aber der
Mensch muss im
Mittelpunkt bleiben.

slashwhy

Kapitel 3 / Abgrenzung

slashwhy

Warum die KI nicht die ganze Arbeit machen sollte

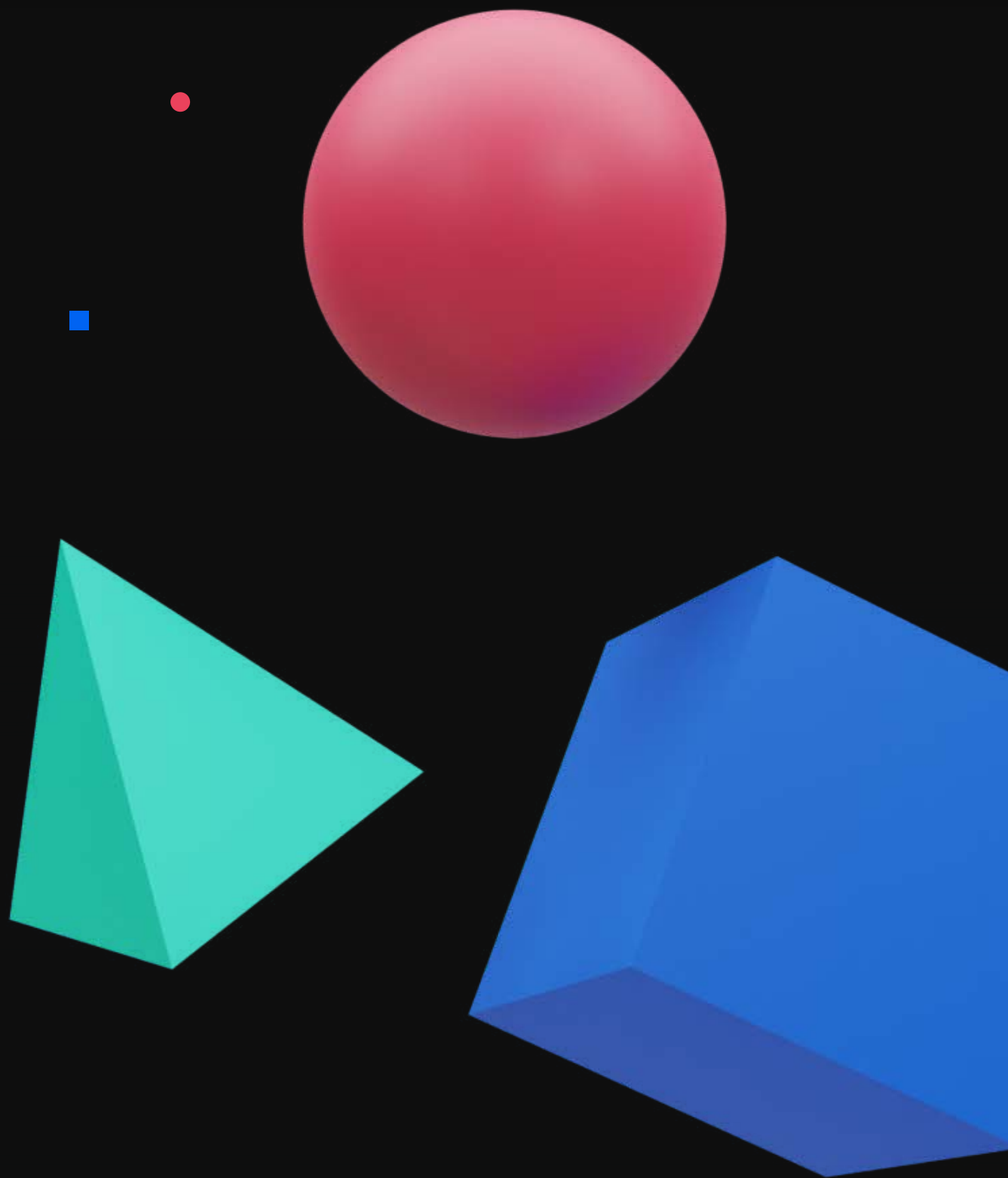
Auch wenn Microsofts CTO Kevin Scott davon spricht, dass in fünf Jahren 95 Prozent des Codes durch KI generiert werden könnten, bleiben offene Fragen. Nicht alles, was technisch möglich ist, ist im Unternehmenskontext sinnvoll. Gerade wenn es um komplexe, langlebige Produkte geht, ist der Unterschied zwischen schnellem Output und tragfähiger Lösung entscheidend.

Bei slashwhy beobachten wir technologische Entwicklungen mit großem Interesse. So auch das Trendthema Vibe Coding. Die Idee dahinter klingt faszinierend: Leute mit wenig oder keiner Code-Erfahrung formulieren Anforderungen in natürlicher Sprache, die KI liefert den passenden Code. Ganze Webseiten, kleine Anwendungen oder Automatisierungen entstehen im privaten und unprofessionellen Umfeld so in wenigen Stunden. Schnell, direkt und scheinbar mühelos.

Wir bei slashwhy sagen klar: Neugierde ist wichtig. Aber Neugierde allein ist keine Grundlage für professionelle Entscheidungen. Deshalb empfehlen wir im professionellen Umfeld, KI-Tools nur kontrolliert und unterstützend einzusetzen, sich aber nicht komplett darauf zu verlassen, geschweige denn ganze Aufgaben an die KI zu übergeben. Eine kritische Gegenüberstellung der beiden Ansätze zur GenAI-Nutzung in der Software-Entwicklung unterstreicht diese Empfehlung.

Vibe Coding vs. AI-Assisted Development

Aspekt	Vibe Coding	AI-Assisted Development
Kurzbeschreibung	Die KI generiert auf Knopfdruck nahezu den gesamten Code. Schnell, aber oft ohne Verständnis, Struktur oder Qualität.	Die KI wird gezielt als Werkzeug für repetitive Aufgaben genutzt. Der Fokus liegt auf Sicherheit und Produkt-Qualität.
Rolle der Entwickler:innen	Anforderung werden formuliert, die KI erstellt den Code	Die KI unterstützt, aber Entscheidungen und Verantwortung bleiben im Team
Codeverständnis	Code wird häufig übernommen, ohne ihn vollständig zu verstehen	Entwickler:innen können den Code nachvollziehen und erklären
Qualitätssicherung	Selten systematisch integriert	Fester Bestandteil des Entwicklungsprozesses (Test, Reviews, Standards)
Teamtransparenz	Geringer Austausch über Vorgehen und Entscheidungen	Gemeinsames Verständnis, dokumentierte Entscheidungen, klare Rollen
Wartbarkeit	Code ist oft inkonsistent und schwer erreichbar	Strukturierter, standardkonformer und langfristig tragfähiger Code
Prozessklarheit	Iteratives Arbeiten ohne definierte Schritte	Klare Regeln, definierte Phasen und Rollenverteilung



Unsere Empfehlung

Die kritischen Fragen, die viele Führungskräfte aktuell rund um AI-Assisted Development stellen, sind kein Zeichen von Skepsis, sondern von Verantwortung. Wer sie stellt, ist nicht der Bremser im System, sondern der Möglichmacher. Denn genau hier liegt der Unterschied zwischen blindem Tool-Einsatz und produktiver, verantwortungsvoller Einführung:

Folgen auf das Hinterfragen auch konkrete Leitplanken?
Oder bleibt es bei Bauchgefühl und Einzelentscheidungen?
Nur wer den Rahmen definiert,

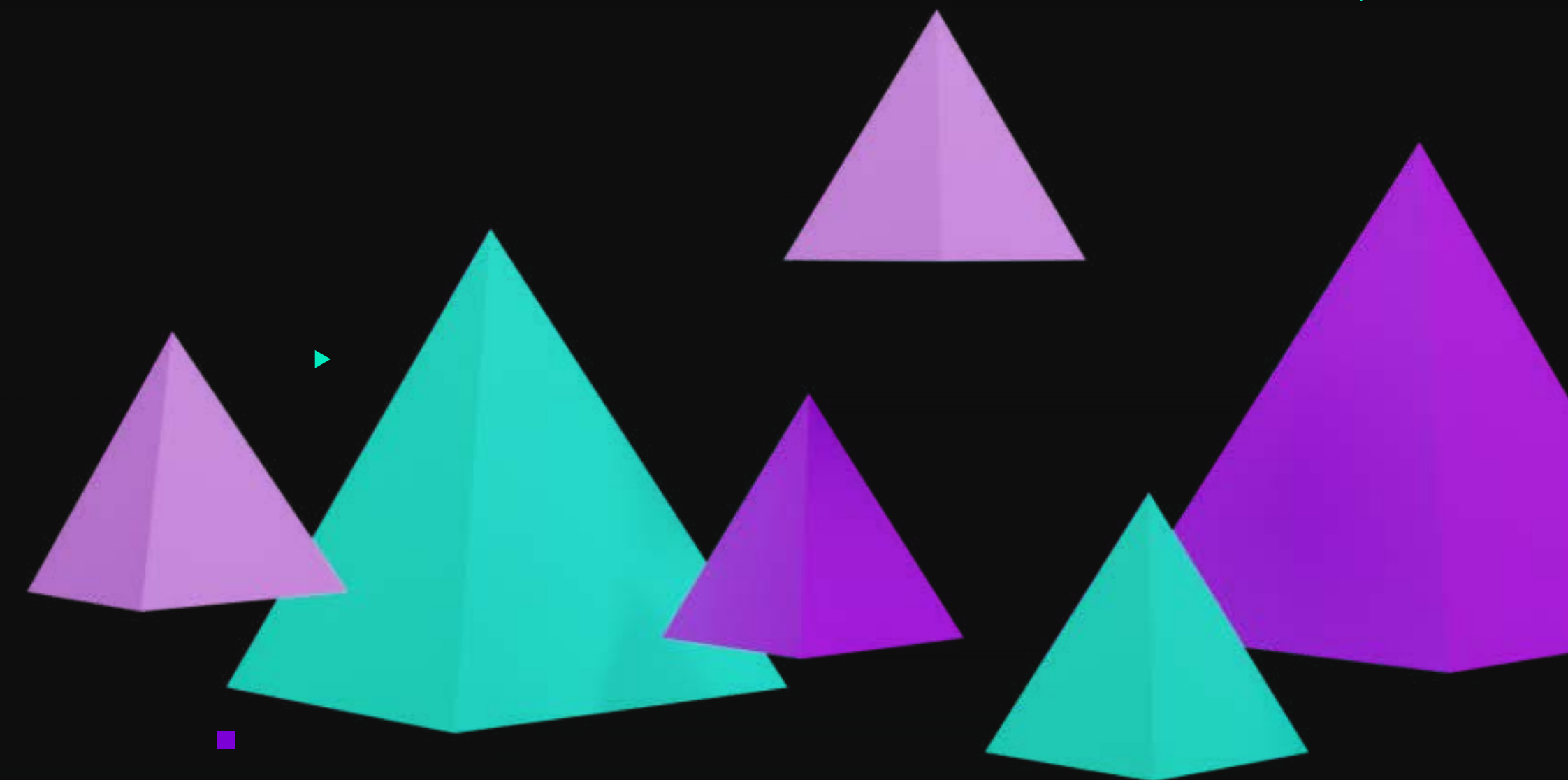
Kapitel 4 / Potenziale

slashwhy

Wo KI in der Software-Entwicklung sinnvoll ist

AI-Assisted Development ist kein Standardprozedere, sondern ein Werkzeug, das gezielt und verantwortungsvoll eingesetzt werden muss. Dort, wo es fachlich passt, vertraglich erlaubt ist und dem Team tatsächlich hilft. Denn gerade in professionellen oder gar regulierten Projekten braucht es dafür klare Rahmenbedingungen und abgestimmte Prozesse.

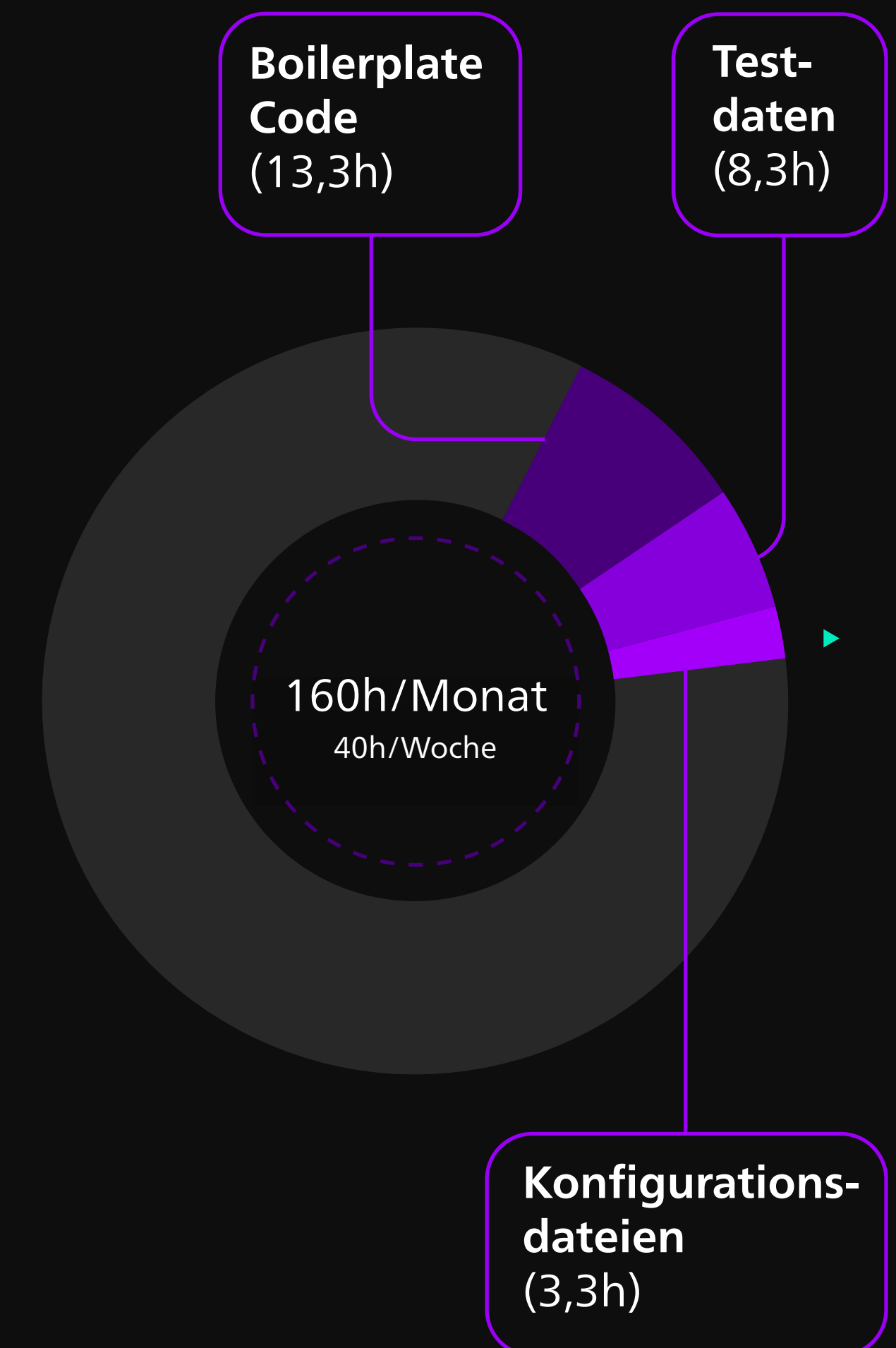
Wo diese Voraussetzungen erfüllt sind, kann KI repetitive Aufgaben vereinfachen und dadurch Freiräume für konzeptionell anspruchsvollere Themen schaffen: Architektur, Qualität, Kommunikation. Besonders wirksam ist das bei standardisierten Aufgaben mit geringem kreativen Spielraum.



Fiktives Rechenbeispiel für Zeit- und Kostenersparnisse

- ✓ Beim Schreiben von **Boilerplate-Code für API-Endpunkte** können Entwickler:innen im Durchschnitt rund 40 Minuten pro Aufgabe sparen. Bei etwa 20 neuen Endpunkten pro Monat ergibt das eine Entlastung von über 13 Stunden.
- ✓ Auch bei der **Generierung von Testdaten** kann KI unterstützen. Statt eine Stunde manuell zu investieren, reichen mit AI-Unterstützung oft 10 Minuten. Bei 10 Testsets pro Monat entstehen so weitere 8,3 Stunden Zeitgewinn.
- ✓ Die **Erstellung einfacher Konfigurationsdateien** lässt sich ebenfalls beschleunigen. Aus durchschnittlich 30 Minuten Aufwand können mit KI etwa 5 Minuten werden. Bei acht Vorgängen im Monat spart das nochmals rund 3,3 Stunden.

Addiert man die aufgeführten Beispiele zusammen, ergibt sich eine **theoretische Zeitersparnis von über 24 Stunden pro Monat und Entwickler:in**. Bei einem marktdurchschnittlichen Stundensatz (gemäß Kununu-Vergleich) von 120 Euro entspricht das einer Entlastung von knapp 2.900 Euro pro Monat. Diese kommt nicht durch reale Einsparungen zustande, sondern durch bessere Verteilung der Ressourcen. Denn diese Arbeitszeit fällt nicht weg, sondern kann gezielt anders genutzt werden. Statt sie in wiederkehrende Aufgaben zu investieren, gewinnen Teams Zeit für nachhaltige Architektur, die Optimierung komplexer Geschäftslogik oder den intensiven Austausch im Team. Also für das, was langfristig Qualität schafft.



Reale Einschätzungen unseres Entwickler:innen-Teams

Neben zahlreichen Agile Coaches, Scrum Mastern, UX & UI Designern und sonstigen Supporting Functions sind bei slashwhy 200 Software-Entwickler:innen beschäftigt.

■ Wir haben ihnen in ihrem Alltag Zeit zum Experimentieren mit GenAI-Tools eingeräumt und nach ihrer kritischen Einschätzung nach einem der verbreitetsten Tools gefragt: GitHub Copilot.

Was gefällt dir?

Unsere Entwickler*innen schätzen an dem Tool vor allem die Beschleunigung der Arbeit und die Minimierung von Schreibarbeit durch die Funktion „Code-Vervollständigung“. Ebenfalls positiv bewertet werden die nahtlose Integration in Entwicklungsumgebungen (IDE) wie VSCode und Rider sowie die Fähigkeit des Tools, auf die Codebasis zuzugreifen. Weitere Pluspunkte sind:

- / Unterstützung bei Refactorings
- / Schnelles Erstellen von Test-Grundlagen und Commit-Nachrichten
- / Generierung von Code-Erklärungen
- / Finden neuer Lösungsansätze, wenn gedanklich festgefahren.

Was gefällt dir nicht?

Ein zentraler Kritikpunkt ist die Fehleranfälligkeit des Tools. Es erfindet häufig Methoden oder Parameter, die nicht existieren, was zu zusätzlichem Aufwand führt. Nutzer berichten, dass die Qualität der Vorschläge inkonsistent ist und teilweise sogar abgenommen hat. Für größere oder klassenübergreifende Aufgaben wird es als ungeeignet angesehen. Weitere negative Aspekte sind:

- / Performance-Probleme und Konflikte mit anderen IDE-internen Tools
- / Prompts müssen sehr präzise formuliert sein
- / Ablenkende und manchmal störende Code-Vorschläge unterbrechen den Arbeitsfluss
- / Technische Hürden, wie das erschwerte Debugging von C# in VSCode
- / Einige empfinden die Antwortqualität als schlechter im Vergleich zu ChatGPT im Browser.

Würde es dir schwer fallen zukünftig ohne das Tool zu arbeiten? Was würdest du vermissen? Warum?

Die Meinungen gehen auseinander: Für viele ist Copilot schon jetzt unverzichtbar, vor allem wegen der nahtlosen IDE-Integration und der blitzschnellen Code-Vervollständigung. Andere bleiben flexibler und setzen parallel weiter auf Google/Gemini oder ChatGPT, weil die Ergebnisse noch nicht immer verlässlich genug sind. Manche tun sich zudem mit Einrichtung und Lernkurve schwer.

Ein Punkt zieht sich aber durch fast alle Antworten: Die Geschwindigkeit beim Schreiben und Refaktorisieren von Code hebt den Alltag auf ein neues Level und schafft Raum für anspruchsvollere Aufgaben. Und genau das würden viele unserer Entwickler:innen am stärksten vermissen.

Kapitel 5 / Tools

slashwhy

Hilfreiche GenAI-Tools

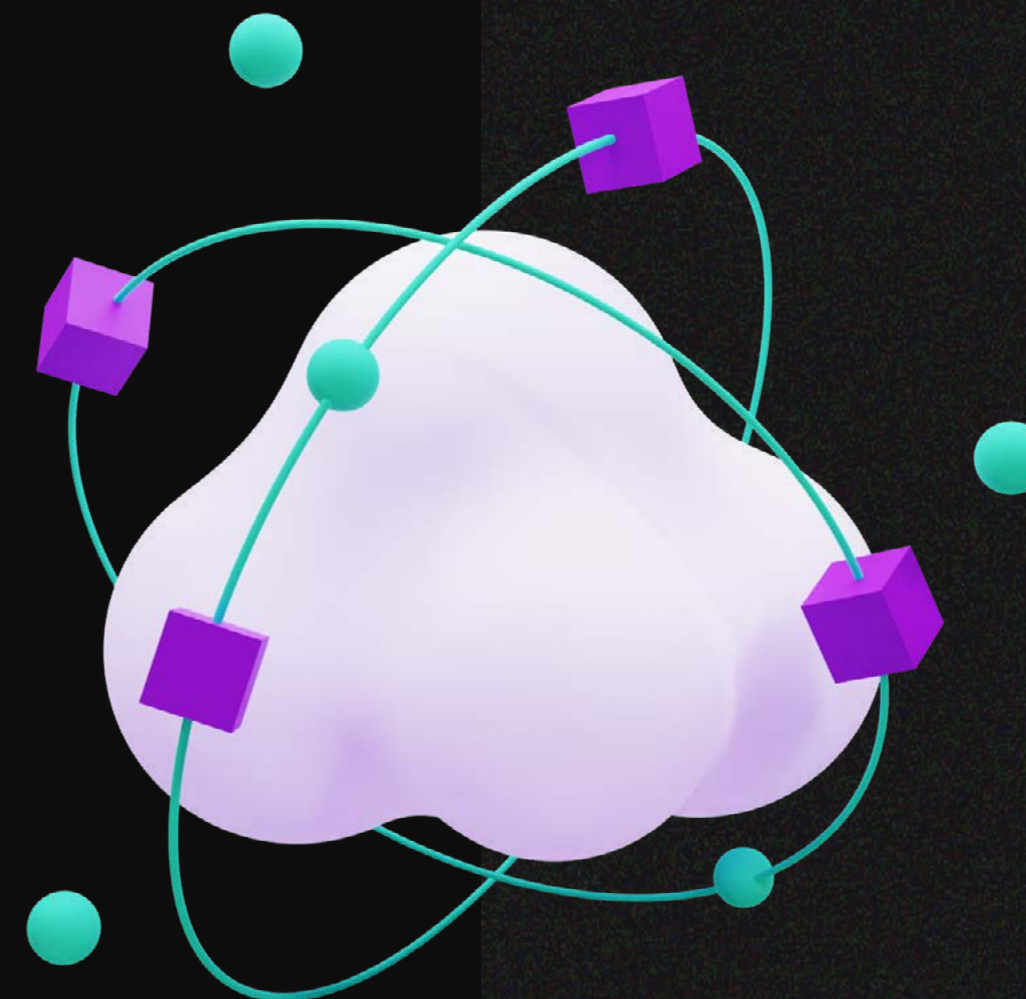
Die Auswahl an GenAI-Tools für die Software-Entwicklung wächst rasant. Doch nicht jedes Tool passt zu jedem Team oder Einsatzzweck. Gerade im Rahmen von AI-Assisted Development ist es wichtig, zwischen kreativen Helfern und verlässlichen Workflow-Komponenten zu unterscheiden.

KI-Tools sollten nicht wahllos eingeführt, sondern gezielt ausgewählt werden. Für den Anfang empfehlen wir, mit einem Tool zu starten, das sich gut in bestehende Prozesse und die Entwicklungsumgebung integrieren lässt. So z. B. GitHub Copilot für wiederkehrende Coding-Aufgaben oder Doku. Tools wie Cursor oder Claude eignen sich gut für exploratives Arbeiten, setzen aber technisches

Verständnis und klare Regeln voraus. Für AWS-nahe Teams kann Amazon CodeWhisperer sinnvoll sein, Google-lastige Teams starten mit Gemini.

Unser Tipp: Teste ein bis zwei Tools im direkten Vergleich, z. B. Copilot + Cursor, oder ChatGPT + Claude und sammle im Team Feedback: Was hilft wirklich? Was stört? Wo entstehen neue Risiken? Was passiert mit den eigenen Daten?

Langfristig ist ein gemeinsamer Werkzeugkasten mit klarer Freigabeliste, Kostenkontrolle und definierten Einsatzbereichen der beste Weg, um aus KI-Tools echten Mehrwert zu schaffen und gleichzeitig einen Wildwuchs an Tools und Prozessen zu vermeiden.



Dein GenAI Quick Start

[Mehr erfahren](#)



**Stand heute ist KI
nicht Ersatz, sondern
maximal Unter-
stützung für deine
Entwickler:innen.**

slashwhy

Zwischenfazit

Die Studien und Beobachtungen der letzten Kapitel verdeutlichen: **Die technische Basis ist längst gelegt. Jetzt braucht es Orientierung, Struktur und aktives Führungsverhalten, um diese Entwicklung zu begleiten.** Wir sehen Führungskräfte in diesem Wandel nicht als Kontrolleure, sondern als Gestalter:innen von Rahmenbedingungen, in denen KI sicher, sinnvoll und im Sinne des Teams genutzt wird. Wir empfehlen Führungskräften von Entwickler:innen-Teams...

- ✓ eine klare Richtlinie für den Einsatz von KI-Tools zu definieren
- ✓ Verantwortlichkeiten festzulegen
- ✓ Kompetenzen im Team systematisch aufzubauen
- ✓ Tool-Landschaften zu konsolidieren, statt auf Einzellösungen zu setzen
- ✓ Qualität, Sicherheit und Nutzerzentrierung frühzeitig mitzudenken

Gerade weil sich die Technologie noch in einem frühen Stadium befindet, haben Führungskräfte die Chance, von Anfang an Rahmenbedingungen zu schaffen, die nicht nur einzelne Tools ermöglichen, sondern nachhaltige und verantwortungsvolle Produktentwicklung mit KI unterstützen. Gerade in Unternehmen ohne klare Guidelines oder Standards wächst ansonsten das Risiko, dass individuelle Vorlieben über die Toolwahl entscheiden und gemeinsame Qualitätsmaßstäbe verloren gehen.

Klare Leitplanken helfen auch, Ängste abzubauen. Wenn von Anfang an definiert ist, welche Aufgaben KI übernehmen darf und welche nicht,

entsteht Vertrauen: im Team, in die Ergebnisse und in die Entscheidung, KI überhaupt einzusetzen. Und erst dann kann AI-Assisted Development sein Potenzial entfalten. Wenn es nicht zufällig, sondern gezielt in den Prozess eingebunden wird. Und das ist keine Frage der Toolwahl, sondern eine Führungsaufgabe.

Gerade bei sicherheitsrelevanten Themen, bei langfristig tragfähigen Architekturen oder in der Konzeption neuer Systemteile ist KI nicht Ersatz, sondern maximal Unterstützung für deine Entwickler:innen. Die Verantwortung für Struktur, Sicherheit und Wartbarkeit sehen wir weiterhin bei den Menschen im Team.

Kapitel 6 / Tipps

slashwhy

Wie Führungskräfte den Einstieg in „AI-Assisted Development“ erfolgreich begleiten

1. Stelle sicher, dass Entwickler:innen den generierten Code wirklich verstehen

Code darf nicht nur funktionieren, er muss verstanden, überprüft und im Team nachvollziehbar sein. Code Reviews gehören bei uns längst zum Standard, doch mit dem Einsatz von KI gewinnen sie zusätzlich an Bedeutung: Gerade bei generierten Codeabschnitten ist es entscheidend, dass Entwickler:innen erklären und begründen können, was übernommen wird. Das stärkt nicht nur die Qualität, sondern auch das gemeinsame Systemverständnis

Hinweis:

An dieser Stelle ist Fingerspitzengefühl gefragt. Code-Reviews sollten nicht in Mikromanagement abgleiten. Trotz sinnvoller Prüfprozesse brauchen Entwickler:innen den Freiraum, neue Werkzeuge mit Neugier und Verantwortung auszuprobieren.

2. Stärke eigenständiges Denken und fördere die KI-Nutzung proaktiv

KI soll entlasten, nicht das eigenständige Lösen komplexer Probleme verdrängen. Führungskräfte können das fördern, indem sie Formate schaffen, in denen konzeptionelles Denken im Fokus steht. Zum Beispiel bei Architektur-entscheidungen oder bei der Bewertung mehrerer Lösungswege. Auch konkrete Schulungen oder Weiterbildungen im Sinne von „Wie arbeite ich mit der KI im Team?“ oder „Welches LLM nutze ich für welche Aufgabe?“ können helfen, Fähigkeiten des Menschen und Computers zielführend miteinander zu verknüpfen.

Hinweis:

Es hilft, Aufgaben bewusst so zu formulieren, dass die KI maximal Assistenz bietet, die Entscheidung aber beim Team liegt. So bleibt Verantwortung erhalten und Kompetenzentwicklung findet weiterhin statt. Zum Beispiel: „Lass dir von der KI mögliche Teststrategien für das Feature vorschlagen und wählt im Team die passende aus.“

3. Definiere Standards für den Einsatz von AI-Assisted Development

Neben klassischen Qualitätsvorgaben braucht es neue Regeln für den KI-Einsatz. Dazu gehören etwa die Benennung der Tools sowie klare Kriterien, welche Aufgaben mit KI umgesetzt werden dürfen und welche nicht.

Hinweis:

Standards sollten praxisnah bleiben. Zu viele Vorschriften können den produktiven Einsatz der Technologie blockieren. Besser ist es, mit einem sinnvollen Rahmen zu starten und diesen auf Basis der gesammelten Erfahrungen gemeinsam weiterzuentwickeln..

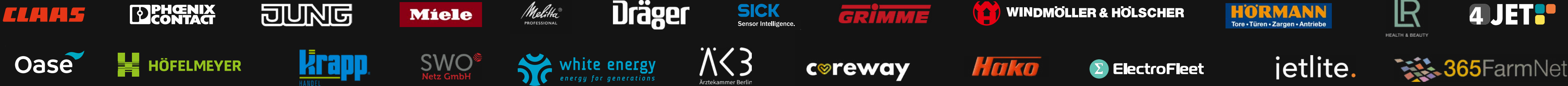
4. Denke Sicherheit und Datenschutz konsequent mit

Beim Einsatz externer KI-Dienste muss klar geregelt sein, wie mit sensiblen Daten umgegangen wird. Prompting Guidelines, abgestimmt mit Datenschutz- und Rechtsteams, schaffen Sicherheit für alle Beteiligten.

Hinweis:

Transparenz ist hier entscheidend. Teams sollten genau wissen, welche Tools genutzt werden dürfen, welche Informationen verarbeitet werden können und wo klare Grenzen gelten. Nur dann entsteht Vertrauen in den kontrollierten Einsatz von GenAI.

Auswahl unserer Kunden:



Fazit

slashwhy

Führungskräfte müssen KI-Nutzung aktiv begleiten

AI-Assisted Development ist kein Selbstläufer. Es funktioniert dann am besten, wenn Menschen, Prozesse und Technologie bewusst aufeinander abgestimmt sind. Führungskräfte spielen dabei eine Schlüsselrolle. Nicht durch Kontrolle, sondern durch Klarheit.

Wer Standards schafft, Freiräume definiert, Wissen im Team stärkt und Verantwortung sichtbar macht, sorgt dafür, dass KI nicht zu Verunsicherung führt, sondern zu besseren Ergebnissen. Dabei geht es **nicht um starre Regeln, sondern um gemeinsames Lernen im Team**, um neue Fragen von

Qualität, Verantwortung und Zusammenarbeit zu beantworten. Gute Führung erkennt: **KI ersetzt nicht das Denken, sie fordert es heraus.** Und genau darin liegt ihre Chance: Menschen mehr Raum für das geben, was wirklich zählt. Und Technologie als Werkzeug benutzen, um schneller, strukturierter und besser dorthin zu kommen.

Der Gedanke, Software mit KI einfach per Sprache entstehen zu lassen, klingt und ist verlockend. Aber genau deshalb braucht es einen klaren Blick. **Denn nur weil etwas beeindruckend**

schnell funktioniert, ist es nicht automatisch der richtige Weg. Vor allem dann nicht, wenn du langfristig tragfähige, sichere und wartbare Produkte entwickeln willst. Wir sehen Vibe Coding als spannendes Experimentierfeld. Für uns hat es in professionellen Projekten aber keinen Platz. Stattdessen setzen wir auf **AI-Assisted Development: eine Arbeitsweise, in der KI gezielt unterstützt, aber nicht entscheidet.** In der Qualität nicht automatisiert, sondern verantwortet wird. Und in der Teams gemeinsam lernen, was gut funktioniert und was nicht.

Wenn du in deinem Unternehmen über den Einsatz von KI in der Entwicklung nachdenkst, stell dir weniger die Frage, wie schnell du damit wirst. Stell dir die Frage, wie bewusst du dabei bleiben willst. Denn Tempo ist kein Selbstzweck. Und: Verantwortung lässt sich nicht prompten.

So kommst du in die Umsetzung

slashwhy

AI-Driven Solutions

Generative KI ist für uns kein Selbstzweck. Sie ist ein Werkzeug, das Menschen entlastet, Produkte verbessert und Prozesse unterstützt – wenn sie sinnvoll eingesetzt und eingebunden wird. An dieser Stelle beraten wir nicht von außen, sondern entwickeln mit dir gemeinsam: von der Idee über den ersten Prototypen bis zur Integration in deine Systeme und Produkte.



Warum mit slashwhy arbeiten

Wir betrachten GenAI nicht als eigenständige Disziplin, sondern als Teil einer ganzheitlichen Produkt- und Systementwicklung. Sie entfaltet ihren Wert erst dann, wenn sie sinnvoll mit Architektur, Daten und Nutzererfahrung zusammenspielt. Genau dafür bringen wir als Entwicklungspartner unsere Erfahrung aus **Embedded, IoT, Cloud und UX** mit in jedes Projekt.

Dieses Know-how erlaubt es uns, KI nicht als loses Add-on aufzusetzen, sondern passgenau in deine Produkte und Systeme zu integrieren.



Make people happy

Software soll das Leben einfacher machen, nicht komplizierter. Für uns bedeutet das: Jede Lösung muss Menschen konkret helfen, sei es deinem Team im Alltag, deinen Usern bei der Nutzung deines Produkts oder ganzen Organisationen in ihren Prozessen. GenAI darf dabei kein Gimmick sein, sondern muss spürbaren Mehrwert bringen und Sinn erfüllen. Nur wenn ein Einsatz am Ende tatsächlich "happy macht" setzen wir ihn um.



Sicherheit zuerst

Datenschutz, Compliance und rechtliche Rahmenbedingungen sind für uns keine nachträgliche Pflichtübung, sondern die Grundlage jedes Projekts. Wir wissen, wie groß die Verantwortung beim Umgang mit Daten und GenAI ist. Deshalb achten wir schon in der Konzeption darauf, dass deine Lösung nicht nur technisch funktioniert, sondern auch den regulatorischen Anforderungen entspricht und Vertrauen schafft.



Langfristig denken

Wir entwickeln keine kurzfristigen GenAI-Experimente, die nach wenigen Monaten wieder verpuffen. Wir schaffen professionelle Lösungen, die sich weiterentwickeln lassen, die auch morgen noch Bestand haben und die sich in deine System-Landschaft einfügen. Dazu gehört, klein zu starten, Ergebnisse zu messen und Schritt für Schritt auszubauen. So entstehen GenAI-Anwendungen, die technologisch und wirtschaftlich nachhaltig sind.

Wo du uns findest



Osnabrück

☎ +49 541 9610445-0
✉ info-os@slashwhy.de
📍 Netter Platz 2
📍 49090 Osnabrück



Hamburg

☎ +49 40 3018722-0
✉ info-hh@slashwhy.de
📍 Ludwig-Erhard-Straße 22
📍 20459 Hamburg

Was ist sinnvoll &
wo fangen wir an? ■
Finden wir es
gemeinsam heraus.



Dr. Joachim Wilharm
Managing Director

☎ Osnabrück
✉ jwilharm@slashwhy.de
📍 joachimwilharm

This work is licensed under the Creative Commons
CC BY-ND 4.0 DEED Attribution-NoDerivs 4.0
International License. To view a copy of this license,
visit <https://creativecommons.org/licenses/by-nd/4.0/> or send a letter to Creative Commons,
PO Box 1866, Mountain View, CA 94042, USA.

slashwhy GmbH & Co. KG
Netter Platz 2 | 49090 Osnabrück

+49 541 96104-0
info@slashwhy.de
www.slashwhy.de