
Softwareentwicklung am Scheideweg

Wie KI-Agenten die Arbeit von Teams verändern — und was das für Visionäre, Agilität und die Zukunft der Softwareentwicklung bedeutet

Executive Summary

Die Softwareentwicklung verändert sich. Nicht auf der Ebene der Methoden: bei der Frage, wer Software entwickelt und wie.

KI-gestützte Entwicklungsumgebungen ermöglichen es heute, spezialisierte Agenten für Architektur, Backend, Frontend, Testing und DevOps zu orchestrieren, koordiniert durch eine einzige Person. Was bislang ein Team aus zahlreichen Spezialisten erforderte, ist für wenige erfahrene Einzelpersonen (mitunter sogar nur einer) mit dem richtigen Setup erreichbar.

Das hat Folgen: für Teamstrukturen, für agile Methoden, für die Frage, wer künftig Software baut. Und vor allem: wer die Entscheidungen trifft.

1. Der Status quo: Agile Softwareentwicklung heute

Seit den frühen 2000er Jahren hat sich agile Softwareentwicklung als dominierendes Paradigma etabliert. Das Agile Manifesto von 2001 war eine Reaktion auf schwerfällige Wasserfallprozesse: zu viel Planung, zu wenig Lieferung, zu lange zwischen Idee und funktionierender Software.

Scrum, Kanban, SAFe und ihre Derivate haben seitdem Millionen von Entwicklungsteams geformt. Schnellere Lieferzyklen, engere Zusammenarbeit mit dem Kunden, kontinuierliche Verbesserung: in vielen Projekten wurden diese Versprechen eingelöst.

Doch mit wachsender Verbreitung wurden auch die Schattenseiten sichtbar:

Zeremonien fressen Zeit. Daily Standups, Sprint Plannings, Reviews, Retrospektiven, Refinements... In einem typischen Scrum-Team verbringt ein Entwickler mehrere

Stunden pro Woche in Meetings, die den Code nicht schneller machen. In großen Organisationen potenziert sich das durch mehrstufige Koordination.

Teams werden groß und schwerfällig. Ein vollständiges agiles Team (Product Owner, Scrum Master, mehrere Entwickler, QA, DevOps, UX) erfordert Abstimmung auf vielen Ebenen. Entscheidungswege verlängern sich. Der ursprüngliche Agilitätsgedanke wird durch schiere Teamgröße konterkariert.

Die Lücke zwischen Vision und Umsetzung bleibt. Wer eine Softwareidee hat, ist selten derjenige, der sie baut. Zwischen dem Visionär und dem fertigen Produkt stehen Backlogs, Priorisierungsgespräche, Sprints, Reviews. Die Reibung ist systemimmanent.

Fachkräftemangel verschärft die Lage. Qualifizierte Softwareentwickler sind knapp und teuer. Kleine Unternehmen, Startups und Organisationen mit begrenztem Budget können sich vollständige Entwicklungsteams schlicht nicht leisten. Ihre Ideen bleiben Ideen.

Diese strukturellen Spannungen sind nicht neu. Neu ist, dass sich eine technologische Antwort abzeichnet: und die ist radikaler als erwartet.

2. Das neue Paradigma: KI als Entwicklungsteam(unterstützung)

Die öffentliche Wahrnehmung von KI in der Softwareentwicklung hat sich in den vergangenen zwei Jahren verschoben. Was als intelligente Autovervollständigung begann (GitHub Copilot, TabNine und ähnliche Werkzeuge) ist heute etwas qualitativ anderes.

Moderne KI-Systeme können nicht mehr nur einzelne Zeilen oder Funktionen vorschlagen. Sie analysieren Anforderungen, entwerfen Architekturen, schreiben Code, generieren Tests, orchestrieren Deployments, in koordinierten Abläufen, die einer arbeitsteiligen Teamstruktur ähneln.

Der Schritt, der das ändert, ist der Übergang vom Werkzeug zum Agenten.

Ein KI-Werkzeug antwortet auf eine Anfrage. Ein KI-Agent verfolgt ein Ziel, plant Zwischenschritte, koordiniert mit anderen Agenten und handelt selbstständig: innerhalb definierter Grenzen, mit menschlicher Aufsicht an kritischen Punkten.

Was heute möglich ist: Konzepte wie das KI-Entwicklungs-OS zeigen, wohin diese Entwicklung führt. Spezialisierte KI-Agenten übernehmen die Rollen eines vollständigen Entwicklungsteams, von Architektur und Backend über Frontend und UX bis zu Testing, Code Review und DevOps. Im Extremfall gibt ein menschlicher Orchestrator die Richtung vor, trifft Grundsatzentscheidungen und nimmt Ergebnisse ab.

Das System arbeitet nicht nach festen Sprints. Aufgaben entstehen und werden bearbeitet, wenn sie gebraucht werden. Entscheidungen fallen dort, wo Expertise sitzt: beim zuständigen Agenten, mit menschlicher Freigabe an den kritischen Stellen.

Ein erfahrener Entwickler oder technisch versierter Produktverantwortlicher kann mit KI-Unterstützung bei klar definierten Aufgabentypen merklich mehr bewegen und Aufgaben übernehmen, für die früher mehrere Spezialisten nötig waren. Für komplexe Architekturentscheidungen, tiefes Domänenwissen und kritisches Review bleibt menschliche Expertise unverzichtbar.

Diese Entwicklung ist im Gange. Für bestimmte Aufgabenbereiche ist sie bereits Praxis, und der Abstand zu dem, was heute noch als Grenze gilt, schrumpft schneller als die meisten Roadmaps einkalkuliert haben.

3. Die Machtverschiebung: Der Visionär gestaltet mit

In der traditionellen Softwareentwicklung gibt es eine Grundspannung: Die Person, die weiß, was gebaut werden soll, kann es meistens nicht selbst bauen. Der Product Owner schreibt User Stories: die Entwickler entscheiden, wie sie umgesetzt werden. Der Gründer hat die Vision, das Team entscheidet über die technische Realisierung.

Das ist keine Wissensfrage. Es ist eine Machtfrage.

Wer den Code schreibt, trifft tausend kleine Entscheidungen, die kein Backlog-Item erfassen kann. Welche Bibliothek wird eingesetzt? Wie wird das Datenmodell strukturiert? Wo werden Abkürzungen genommen, weil der Sprinttermin naht? Die technische Umsetzung formt das Produkt: oft stärker als jede Roadmap.

KI-gestützte Entwicklungsumgebungen beginnen, das zu ändern.

Ein Visionär, der früher auf ein Entwicklerteam angewiesen war, kann heute zunehmend selbst den Takt vorgeben. Er muss keinen Code schreiben. Er muss aber wissen, was er will, und das präzise kommunizieren können. Die Qualität des Ergebnisses hängt weniger von technischem Detailwissen ab als von konzeptioneller Klarheit.

Das verändert die Teamstrukturen:

Kleinere Teams mit mehr Spielraum. Ein erfahrener Produktdenker mit KI-Unterstützung kann in bestimmten Bereichen Aufgaben übernehmen, für die bislang mehrere Spezialisten nötig waren.

Neue Rollenbilder. Der klassische „Full-Stack-Entwickler“ war ein Versuch, Breite mit Tiefe zu verbinden. Was jetzt entsteht, ist anders: der technisch informierte Visionär, der KI-Agenten führt statt selbst zu programmieren. Konzeptionelle Kompetenz wird zum Differenzierungsmerkmal.

Technisches Wissen bleibt wertvoll, aber anders. Wer versteht, wie Systeme funktionieren, kann KI-Agenten besser steuern, Vorschläge kritisch bewerten und Grenzen erkennen. Das Spezialwissen des Entwicklers wird nicht wertlos. Es verändert seinen Einsatzort. Weniger Ausführung, mehr Urteilsvermögen.

Der Zugang öffnet sich. Startups, kleine Unternehmen, Einzelpersonen mit einer guten Idee und begrenztem Kapital bekommen Zugang zu Entwicklungskapazität, die bislang gut finanzierten Organisationen vorbehalten war. Das verschiebt die Kräfteverhältnisse in vielen Bereichen. Nicht irgendwann: jetzt.

Diese Verschiebung ist im Gange. Wer sie aktiv gestaltet, baut einen Vorsprung auf, den Nachzügler nur schwer aufholen.

4. Was wird aus der Agilität?

Scrum ist nicht tot. Aber es steht von einer unerwarteten Seite unter Druck.

Die agile Bewegung entstand als Reaktion auf starre Prozesse. Ihr Kern war nie Scrum oder Kanban. Ihr Kern war das Prinzip: Liefere früh, liefere oft, lerne aus dem Feedback, passe dich an. Die Zeremonien waren Mittel zum Zweck, keine Selbstzwecke.

Mit KI-gestützten Entwicklungsumgebungen tritt dieser Kern in anderer Form wieder hervor.

Sprints weichen kontinuierlichem Fluss. Wenn ein KI-Team in Stunden liefert statt in Wochen, ergibt die künstliche Bündelung von Arbeit in zweiwöchige Zyklen wenig Sinn. Im Takt des tatsächlichen Bedarfs entstehen Aufgaben, werden priorisiert, bearbeitet und abgenommen.

Zeremonien weichen adaptiver Koordination. Das Daily Standup existiert, weil Menschen sich abstimmen müssen. KI-Agenten stimmen sich kontinuierlich ab. Ihr Status ist jederzeit abrufbar. Was bleibt, ist die menschliche Entscheidung: Wo geht es als nächstes hin? Was ist wirklich wichtig?

Das agile Grundprinzip überlebt, in reinerer Form. Kundenmehrwert vor Prozesstreue. Funktionsfähige Software vor Dokumentation. Reaktion auf Veränderung vor starrem Plan. Diese Werte verlieren in einer KI-gestützten Umgebung nichts von ihrer Gültigkeit. Sie kommen stärker zum Tragen, weil die Reibung der Umsetzung sinkt.

Die Rolle des Menschen verändert sich: nicht mehr Zeremonienmeister oder Ressourcenplaner. Der Mensch wird zum Visionär und Qualitätshüter. Die Frage „Was wollen wir bauen?“ bleibt menschlich. Die Frage „Wie wird es gebaut?“ zunehmend weniger.

Adaptive Autonomie als Arbeitsmodell. Statt eines fixen Automatisierungsgrads entscheidet der Visionär situativ: Bei vertrauten, gut verstandenen Aufgaben arbeiten die KI-Agenten autonom. Bei neuen, riskanten oder kritischen Entscheidungen wird der Mensch an jedem relevanten Schritt eingebunden. Die Kontrolle bleibt, aber sie wird gezielt eingesetzt.

Das ist keine Abkehr von Agilität. Es ist ihre Weiterentwicklung.

5. Risiken und offene Fragen

Paradigmenwechsel haben Schattenseiten. Das hier ist keine Ausnahme.

Abhängigkeit von wenigen Plattformen. Wer seine Entwicklungsprozesse auf eine KI-Infrastruktur aufbaut, macht sich von deren Verfügbarkeit, Pricing und strategischen Entscheidungen abhängig. Was heute günstig ist, kann morgen teuer sein. Was heute funktioniert, kann sich durch ein API-Update verändern.

Eine Antwort darauf ist der hybride Stack: lokale Sprachmodelle für datensensible oder standardisierte Aufgaben, Cloud-Modelle dort, wo Reasoning-Tiefe und Kontextlänge gefragt sind. Leistungsfähige lokale Modelle (etwa Qwen2.5-Coder oder DeepSeek) können heute auf geeigneter Hardware viele Coding-Aufgaben übernehmen, ohne dass Daten das eigene System verlassen. Für komplexe Architekturentscheidungen bleibt der Cloud-Einsatz sinnvoll. Das funktioniert nur mit klarer Governance: wer nutzt welches System für welchen Aufgabentyp, unter welchen Datenschutzauflagen. Souveränität bedeutet hier nicht zwingend ein geschlossenes lokales System, sondern Kontrolle über die eigene Infrastruktur und Datenstrategie.

Qualität und Verantwortung. KI-Agenten machen Fehler, manchmal subtile, schwer erkennbare. Verantwortlich ist, wer die Freigabe erteilt hat. Das setzt voraus, das Ergebnis beurteilen zu können. Diese Fähigkeit muss aktiv erhalten werden. Sie verkümmert schnell, wenn man aufhört hinzuschauen.

Wissensverflachung. Wenn KI-Agenten Routineaufgaben übernehmen, besteht die Gefahr, dass das Wissen, das in diesen Aufgaben steckt, nicht mehr aufgebaut wird. Der nächste Visionär akzeptiert Architekturentscheidungen vielleicht einfach, ohne zu verstehen, warum sie so getroffen wurden. Organisationales Gedächtnis muss bewusst gepflegt werden.

Die Kreativitätsfrage. KI kombiniert, was sie gelernt hat: überzeugend, aber nicht originell. Neue Ideen, ungewöhnliche Ansätze, das quere Denken, das zu Durchbrüchen führt: das bleibt menschlich. Die eigentliche Gefahr ist nicht, dass KI zu kreativ wird, sondern dass die Bequemlichkeit des KI-Einsatzes die menschliche Kreativität einschläfert.

Soziale Auswirkungen. Wenn ein Visionär mit KI-Team die Arbeit von fünf Entwicklern leistet, stellt sich die Frage, was mit diesen fünf passiert. Neue Rollen

entstehen, alte verschwinden. Das war bei jedem Technologiewandel so. Dieser Übergang ist selten reibungslos, und er wird es diesmal auch nicht sein.

Diese Risiken sind kein Argument für Stillstand. Aber wer sie ignoriert, zahlt irgendwann den Preis.

6. Ausblick: Drei mögliche Zukünfte

Wie jede transformative Technologie entwickelt sich auch KI in der Softwareentwicklung in verschiedene Richtungen: je nachdem, wie Organisationen, Entwickler und Gesellschaft damit umgehen.

Szenario 1: KI als Werkzeug. KI wird zu einem leistungsfähigen Hilfsmittel, vergleichbar mit dem Übergang von der Schreibmaschine zum PC. Entwickler werden produktiver, Teams schlanker. Die Grundstruktur bleibt erkennbar. KI beschleunigt, ersetzt aber nicht.

Dieses Szenario ist auf kurze Sicht das wahrscheinlichste. Es ist komfortabel und riskant für alle, die zu lange warten.

Szenario 2: KI als Kollege. Die Grenze zwischen menschlichem Entwickler und KI-Agent verschwimmt. Teams bestehen aus Menschen und Agenten mit klar definierten Rollen, gegenseitigen Erwartungen, etablierten Zusammenarbeitsmodellen. Der Mensch gibt die Richtung vor und trifft Entscheidungen. Der Agent setzt um und macht Vorschläge.

Dieses Szenario verlangt neue Führungsformen, neue Kompetenzen, neue ethische Leitlinien: und Organisationen, die das aktiv angehen wollen.

Szenario 3: KI als Ersatz. In bestimmten Bereichen (gut definierte, wiederkehrende Entwicklungsaufgaben) übernimmt KI vollständig. Nicht in einem großen Schritt, sondern schrittweise. Was bleibt, sind Strategie, Domänenwissen, konzeptionelle Arbeit.

Das ist kein Worst Case. Für viele Bereiche ist es die konsequente Fortsetzung des aktuellen Trends.

Die Realität wird eine Mischung aus allen drei sein — und wahrscheinlich nicht in der Reihenfolge, die irgendjemand heute voraussagt. Wer jetzt einschätzt, welches Szenario für seinen Kontext relevant ist, ist besser dran als wer wartet.

7. Das KI-Entwicklungs-OS als Antwort

Das, was bisher beschrieben ist, ist kein Gedankenexperiment.

Das KI-Entwicklungs-OS setzt diese Machtverschiebung praktisch um: Ein Solo-Entwickler oder ein kleines Team bekommt durch spezialisierte KI-Agenten die Schlagkraft eines vollständigen agilen Entwicklungsteams: ohne Koordinationsaufwand, ohne starre Sprint-Zyklen, ohne die Reibungsverluste großer Teamstrukturen.

Der Visionär behält die Kontrolle. Das System schlägt vor, fragt nach, wartet auf Freigabe und setzt dann um. Jeder kritische Schritt wird dokumentiert, jede Entscheidung bleibt nachvollziehbar.

Das Ziel ist nicht, Entwickler zu ersetzen, sondern Menschen mit konzeptioneller Klarheit zu befähigen, hochwertige Software zu bauen: und erfahrene Entwickler dort einzusetzen, wo es wirklich zählt: Architekturentscheidungen, kritisches Review, Domänenkompetenz.

Das Konzept ist auf tech-affine Teams und Unternehmen ausgelegt, die Softwareentwicklung aktiv neu denken: nicht als Universallösung, sondern als gezieltes Instrument für die, die es einsetzen können und wollen.

Das Konzept befindet sich in aktiver Entwicklung. Wer neugierig ist — auf die Architektur, auf erste praktische Erfahrungen oder auf Pilotprojekte — kann sich melden.

Verfasser: Alexander Grein, IT Kombinat GmbH

Dieses Whitepaper wurde auf Basis eines laufenden Konzept- und Entwicklungsprojekts verfasst.

Stand: Juni 2026.

Dieser Text wurde mit Unterstützung von KI-Werkzeugen erstellt.