

KI in der Software-Entwicklung

Automatisierung, neue Kompetenzen
und Verantwortung

Dr. Martin Mähler
IBM Wissenschaftsbeziehungen



Der größte Umbruch

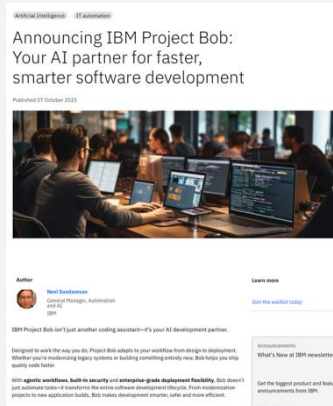
KI beschleunigt und automatisiert
Softwareentwicklung

Assistenz ist Standard: 'Code completion on
steroids':

Agenten & autonome Workflows entstehen

IBM ist sehr aktiv bei der Transformation:

- watsonx code Assistents
- Project Bob



Heute schon im Einsatz

Produktivitätsgewinne sind real (20–80 % je nach
Tool/Use Case)

Automatische Dokumentation & Code-Erklärung
(auch für Legacy-Sprachen wie COBOL)

Code-Übersetzung: Skills werden transferierbarer
(z.B. COBOL $\rightleftharpoons$ Java)

Unterstützung bei Code-Reviews, z.B. auf
Performance

Vorsicht: 'Vibe Coding': von natürlicher Sprache zu
Spezifikation, Design, Tasks & Code

Je Hardware- bzw. „mission critical“ näher die
Programme, umso zurückhaltender/vorsichtiger ist
derzeit der Einsatz

Herausforderungen

Inhaltliche Kompetenz bleibt zentral: Was ist 'guter Code'?

KI unterstützt, ersetzt aber keine vollständigen Reviews

Grenzen der KI erkennen

Neues Arbeitsparadigma: Prompt → Toolchain

Prompting als Schlüsselkompetenz für Qualität & Reviews

Der Developer-Laptop wird zum Tool des LLMs

LLMs orchestrieren passende Tools/Agents statt nur Antworten zu geben

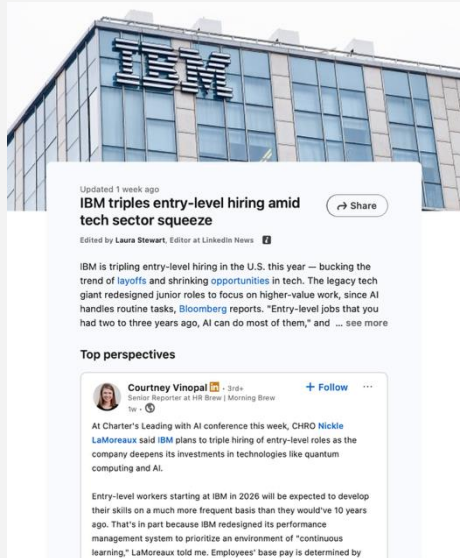
Größere Kontextfenster → tieferes Code-Verständnis

Ausbildung & Karrierepfade: Das Senior-Dilemma

Wegfall klassischer Junior-Aufgaben erschwert die Lernpfade

Analogie: erst selbst feilen lernen, dann dazu Maschinen nutzen

Wer heute nicht in Juniors investiert, könnte später ein größeres Problem haben



Motivation & Human Quality

Risiko: Unter-Nutzung von KI, weil viele „einfach gerne selbst“ coden

Freude am Entwickeln erhalten; Entfremdung vermeiden

Balance: Automatisierung dort, wo es entlastet; Kreativität erhalten

01

LLMs are most helpful on programming problems that are

- Specific
- Limited in scope
- Focused on a programming language with a robust sharing history

02

LLMs can help us identify implementation options, but

- So far, they don't provide an adequate substitute for judgment
- Subject matter expertise and contextual expertise still matter in decision-making

03

The nature of LLMs supports reproducing existing logic

- The more innovation – or “newness” – the approach required, the less these tools can help
- Conveniently that describes few of the problems we face professionally

Neue Kernkompetenzen für Developer

Spezifikationen & Requirements präzise formulieren

'Big Picture' & Systemarchitektur statt reine Zeilenproduktion

Validierung/Bewertung von KI-Ergebnissen
(Korrektheit, Sicherheit, Datenschutz)

Datenqualität, Governance und sichere Integration in Prozesse

=> KI bietet eine Chance zur persönlichen Entwicklung (Job Enrichment)

=> Attraktiver Karrierepfad für Juniors, die sich darauf einlassen

Trends bis 2027

Agentic AI automatisiert Workflows bereits heute punktuell

Agenten als größerer Umbruch als Microservices

Interaktiver Partner für die Softwareentwicklung in Unternehmen, um den gesamten Lebenszyklus – vom Entwurf bis zur Bereitstellung – zu optimieren

- Verständnis für den Kontext des gesamten Repositories, die Absichten des Entwicklers und der Sicherheitsstandards
- KI-Agenten für Code, Tests, Dokumentation und Pipelines
- Integrierte Sicherheitsfunktionen (Vulnerability/Secrets Detection)



Fazit

Die Rolle wandelt sich: von Coder zum Systemgestalter und KI-Dirigenten

Mehr Verantwortung: Automatisierung, Qualität, Sicherheit, Integration

KI ersetzt nicht Developer – sie verändert aber grundlegend, wie sie entwickeln

Die Grundkompetenz bleibt notwendige Voraussetzung



Bild: <https://obvious.com>