

Hoeveel productiever maakt AI ontwikkelaars?

27 augustus 2025, 12:43

Pascal Pollet

Wat AI echt oplevert voor softwareontwikkeling, volgens een Stanford-studie

Een van de belangrijkste toepassingsgebieden van generatieve AI in softwareontwikkeling is de ondersteuning van programmeurs bij het schrijven van code. Hiervoor bestaan inmiddels tal van gespecialiseerde AI-tools voor codering. Sommige bedrijven beweren zelfs dat zij op korte termijn een deel van hun softwareontwikkelaars zullen kunnen vervangen door AI. Maar wat mag u eigenlijk realistisch verwachten van AI voor programmeurs?

Stanford University onderzocht in een grootschalige studie de impact van AI-ondersteuning op de productiviteit bij het schrijven van softwarecode. Wat deze studie bijzonder maakt, is de gehanteerde onderzoeksmethodologie. In plaats van te werken met kunstmatige testen, baseerden de onderzoekers zich op echte praktijkgegevens van meer dan 600 bedrijven die hen toegang verleenden tot hun code-opslagplaatsen (‘repositories’).

Hierdoor kregen ze inzage in het werk van ruim 100.000 ontwikkelaars en analyseerden ze miljarden regels code. Dit stelde hen in staat om onder meer te meten hoe vaak wijzigingen (‘commits’) werden doorgevoerd in de repositories. De productiviteit werd daarbij niet afgemeten

aan het aantal geschreven regels code, maar aan de functionaliteit van de geleverde software.

Wat levert AI nu écht op voor softwareontwikkeling?

Uit het onderzoek blijkt dat **AI-ondersteunde codering** in eerste instantie de output met ongeveer 30 à 40% verhoogt. Die winst is echter niet volledig blijvend: het gebruik van AI introduceert ook fouten die programmeurs nadien moeten opsporen en corrigeren. Dit extra werk verlaagt de gemiddelde winst uiteindelijk tot ongeveer 15 à 20%.

Wanneer werkt AI wél en wanneer niet?

De studie toont aan dat de impact van **AI in softwareontwikkeling** sterk varieert afhankelijk van verschillende factoren:

1. Complexiteit van de programmeertaak

AI presteert aanzienlijk beter bij eenvoudige programmeertaken. Bij complexere taken zijn de winsten kleiner, en in sommige gevallen kan AI zelfs leiden tot een afname van de productiviteit.

2. Maturiteit van de codebase

AI werkt efficiënter bij nieuwe (‘greenfield’) projecten dan bij bestaande (‘brownfield’) codebases. Enkele concrete cijfers uit het onderzoek:

- Eenvoudige greenfield taken: 30–35% winst
- Complexe greenfield taken: 10–15% winst
- Eenvoudige brownfield taken: 15–20% winst
- Complexe brownfield taken: 5–10% winst

3. Populariteit van de programmeertaal

De effectiviteit van AI hangt ook af van de gebruikte programmeertaal. Voor minder populaire talen, zoals Cobol, levert AI nauwelijks voordelen op, zelfs niet bij eenvoudige taken. Integendeel: bij lage populariteit en hoge complexiteit kan AI de productiviteit zelfs negatief beïnvloeden.

Voor populaire talen zoals **Python** en **Java** daarentegen zijn de winsten wél significant, omdat AI-systemen veel beter getraind zijn op de grote hoeveelheid beschikbare code in deze talen.

4. Omvang van de codebase

Naarmate de omvang van de codebase toeneemt, nemen de voordelen van AI merkbaar af. Dit is relevant, aangezien de meeste moderne codebases uit meer dan enkele duizenden regels code bestaan.

Dit wordt deels verklaard doordat de prestaties van **Large Language Models (LLM's)** voor coderingstaken afnemen naarmate de contextlengte toeneemt. De contextlengte is de maximale hoeveelheid informatie (uitgedrukt in tokens) die een taalmodel tegelijk kan verwerken, zowel uit de invoerprompt als uit de gegenereerde uitvoer.

AI als krachtige, maar geen magische oplossing voor softwareontwikkeling

Generatieve AI maakt het onmiskenbaar mogelijk om de prestaties van softwareontwikkelaars aanzienlijk te verhogen. Toch is nuance op zijn plaats. **Productiviteitsstijgingen** zijn in de softwarewereld geen nieuw fenomeen. De afgelopen decennia zijn vaak nog grotere productiviteitswinsten gerealiseerd door over te schakelen op nieuwere programmeertalen. Zo leveren scriptingtalen zoals Python een productiviteitsvoordeel op van ongeveer 30% tot 70% ten opzichte van procedurele talen zoals Fortran of C++. Vergeleken met Assembly zijn deze moderne talen minstens een factor 5 sneller in termen van ontwikkeltijd.

Generatieve AI is dus een krachtige accelerator voor softwareontwikkeling, maar geen universeel recept voor hogere efficiëntie. De context van de programmeertaal tot de aard van de taak bepaalt in grote mate het succes. Voor organisaties is het dus zaak om **AI slim te integreren in de ontwikkelpraktijk**, eerder dan te rekenen op 'magische' prestatiesprongen.

Bronnen

- Https://www.youtube.com/watch?v=tbDDYKRFjhk
- Softwareengineeringproductivity.stanford.edu

Auteurs



Pascal Pollet