

Willkommen zurück im software engineering

Alisa Lafoks



Gute Nachricht: Wir werden wieder zu Software-Ingenieuren. Und nicht zu Codern.

Software engineering ist das Lösen von Problemen. Wie in der Mathematik: Es gibt ein Problem — es muss gelöst werden. Womit du es löst, in welcher Sprache, mit welchem Werkzeug — ist im Grunde egal. Wichtig ist das Ergebnis: Das Problem ist gelöst oder nicht.

Genau das war das Interessante. Sich hinsetzen und denken. Eine Lösung finden, die es vor dir nicht gab. Etwas Neues ausdenken und zum Laufen bringen. Intellektuelle und kreative Arbeit — dafür sind wir in diesen Beruf gegangen. Und nicht dafür, einen halben Tag mit einer Config zu kämpfen, eine Woche mit einem Framework, Monate mit der Infrastruktur.

Inhaltsverzeichnis

Wie Ingenieure zu Codern wurden	2
Brooks hat es schon 1986 benannt	2
AI übernimmt die Drecksarbeit	2
Willkommen zurück	3

Wie Ingenieure zu Codern wurden

Und dann, mit der Zeit, wurde software engineering unmerklich zu Coding. Nicht weil die Ingenieure dümmer geworden wären — sondern weil sich Frameworks und Infrastruktur ins Unanständige vermehrt haben. Jede Bibliothek muss man kennen, jeden Service anbinden, jede Build-Pipeline einrichten.

Und die Ingenieure versanken. Versanken im Coding und im Einrichten von allem Möglichen: Umgebungen hochziehen, sich mit dem Deployment herumschlagen, den Build reparieren, Abhängigkeiten aktualisieren, googeln, warum Version A nicht zu Version B passt. Versanken darin — statt die intellektuellen Probleme zu lösen, um die es in unserem Beruf eigentlich geht. Für Kreativität blieb ihnen weder Kraft noch Aufmerksamkeit.

So kam es, dass aus dem Ingenieur ein Coder wurde. Ein Mensch, der den ganzen Tag Code rund um das Problem schreibt und kaum zum Problem selbst vordringt. Probleme zu lösen wurde zu einem seltenen, fast festlichen Zustand — eine kurze Pause zwischen zwei Portionen Coding.

Brooks hat es schon 1986 benannt

Das hat einen Namen. Frederick Brooks teilte die Komplexität von Software in zwei Arten: die essenzielle und die akzidentelle.^[1]

Die essenzielle ist die Schwierigkeit des Problems selbst. Sie lässt sich nicht entfernen: Sie ist das Engineering selbst — genau diese intellektuellen Probleme. Die akzidentelle ist alles, was drumherum gewachsen ist: Sprachen, Frameworks, Infrastruktur, Verdrahtung, Werkzeuge. Coding, nur damit das alles zusammenpasst.

Brooks warnte: Es gibt keine Silberkugel, akzidentelle Komplexität lässt sich nur zurückschneiden. Dreißig Jahre lang hat sie niemand zurückgeschnitten — man hat sie ausgebaut. Jedes neue Framework versprach, das Leben einfacher zu machen, und fügte eine weitere Schicht Code hinzu, die man jetzt auch noch schreiben und im Kopf behalten muss. So hat Coding das Engineering aufgefressen.

AI übernimmt die Drecksarbeit

Zum ersten Mal gibt es jemanden, dem man genau dieses Coding überlassen kann.

AI übernimmt genau das, was zu schreiben langweilig ist: Code, Boilerplate, Daten von hier nach da schaufeln, Glue-Code, Configs, „wie mache ich das hier in diesem Framework“. Die öde Arbeit des Codeschreibens — eben das Coding. Die Drecksarbeit, in der wir die letzten Jahrzehnte versunken sind. Was Brooks accidental nannte, macht die Maschine schneller und ohne Gejammer.

Viele macht das Angst: „AI ersetzt die Ingenieure.“ Tja — sie fürchten das Falsche. AI ersetzt nicht den Ingenieur. Sie ersetzt den Coder — den Teil unseres Tages, der nie Engineering war.

Willkommen zurück

Was bleibt also uns, den Programmierern? Das Essenzielle bleibt. Entscheiden, was wir überhaupt bauen und wozu. Das Problem verstehen. Die Form wählen. Entscheidungen treffen, für die man später geradesteht. Das Schwere und Interessante, genau diese intellektuellen Probleme, für die wir hergekommen sind.

Und jetzt können wir, Software-Ingenieure, endlich vom Coding zum Kern von software engineering zurückkehren — zum Lösen von Problemen. Vom Coder zurück zum Ingenieur.

Das ist echtes software engineering. Kein Kampf mit Werkzeugen — Probleme lösen. Wie in der Mathematik. Wie von Anfang an.

Also: willkommen zurück. Der Coder ist jetzt die Maschine — und wir sind wieder Ingenieure.

Und ehrlich gesagt — das Leben ist wieder interessant geworden.

[1] Frederick Brooks, „No Silver Bullet — Essence and Accident in Software Engineering“ (1986). Die essenzielle (essential) Komplexität ist die Schwierigkeit des Problems selbst; die akzidentelle (accidental) ist die Schwierigkeit der Werkzeuge, mit denen wir es lösen. https://en.wikipedia.org/wiki/No_Silver_Bullet