



Kollege Roboter

Die Nachfrage nach Software ist enorm. Allein in unserer Firma, schätze ich, würden wir locker noch zehn Entwickler benötigen, um die Wünsche und Anforderungen umzusetzen, die derzeit bestehen.

Aber woher nehmen? Und selbst wenn sie gefunden sind: Gehalt, Freizeit, Urlaub und Betriebsausflüge wollen diese Mitarbeiter selbstverständlich auch. Welch Lichtblick ist da die neue KI von OpenAI (<https://openai.com>), die aus natürlicher Sprache Code erzeugt.

Sie lachen? Weil Sie wissen, was jetzt kommt? Wie oft schon wurde über Programmgeneratoren und Software Factories fabuliert? Wie viele dieser Programme wurden entwickelt, mit hohem Marketingaufwand positioniert, um dann in der Versenkung zu verschwinden? In der Breite durchgesetzt haben sie sich auf jeden Fall nie. Das hat verschiedene Gründe, auf die ich jetzt gar nicht eingehen will.

Doch mit der KI von OpenAI ist die Situation anders: Sie formulieren: „Baue eine Software, die über eine Oberfläche Personen in einer Datenbank verwalten kann.“ Kaum gesprochen, rattert die KI los und spuckt nach überschaubarer Zeit Code aus, der diese Anforderungen erfüllen soll.

Hier kommt sicher Ihr Einwand: Woher weiß die KI, welche Daten für eine Person nötig sind? Antwort: Einen großen Teil hat die KI durch Studium von vorhandenen Lösungen erlernt. Vor- und Nachname, Straße, PLZ und Ort beherrscht sie wahrscheinlich aus dem Effeff. Weitere Felder ließen sich einfach hinzufügen: „Füge der Person noch ein GEZ-bezahlt-Feld hinzu.“

Je mehr Anwendungen entstünden, umso mehr wüsste die KI und könnte dadurch wiederum hilfreicher sein.

Sie müssten keinen Datenbankzugriffslayer mehr schreiben, kein ViewModel mehr programmieren und auch kein Formular gestalten.

Stellt sich nur die Frage, wie der Code dann aussähe. Würde er den Clean-Code-Prinzipien genügen? Wäre er überhaupt lesbar? Aber nachdem das Grundmaterial von bestehender Software kommt, könnten die Clean-Code-Prinzipien ebenfalls in den generierten Code eingehen.

Freilich lässt sich aus einem Demovideo nicht darauf schließen, was das Ding wirklich drauf hat. Außerdem sind die Anforderungen in der Welt so unterschiedlich, dass es unwahrscheinlich erscheint, dass eine KI das alles leisten kann.

Aber statt eines Web-Baukastens ein System zu haben, das sogar auf gesprochene Anforderungen hin eine Software produziert, ist schon prickelnd. Vor allem wenn man so langsam tippt, wie ich das tue.

Viel Spaß mit der dotnetpro

Tilman Börner
Chefredakteur dotnetpro



Ralf Westphal

setzt sich damit auseinander, was den Wert einer Software ausmacht (S. 8)



Uwe Vigenschow

erörtert, wie sich agiles Vorgehen an die Problemstellung anpassen kann (S. 28)



Michael Wresche

erklärt, wie Sie statische Codeanalyse in eine Continuous-Delivery-Umgebung einbauen (S. 72)