

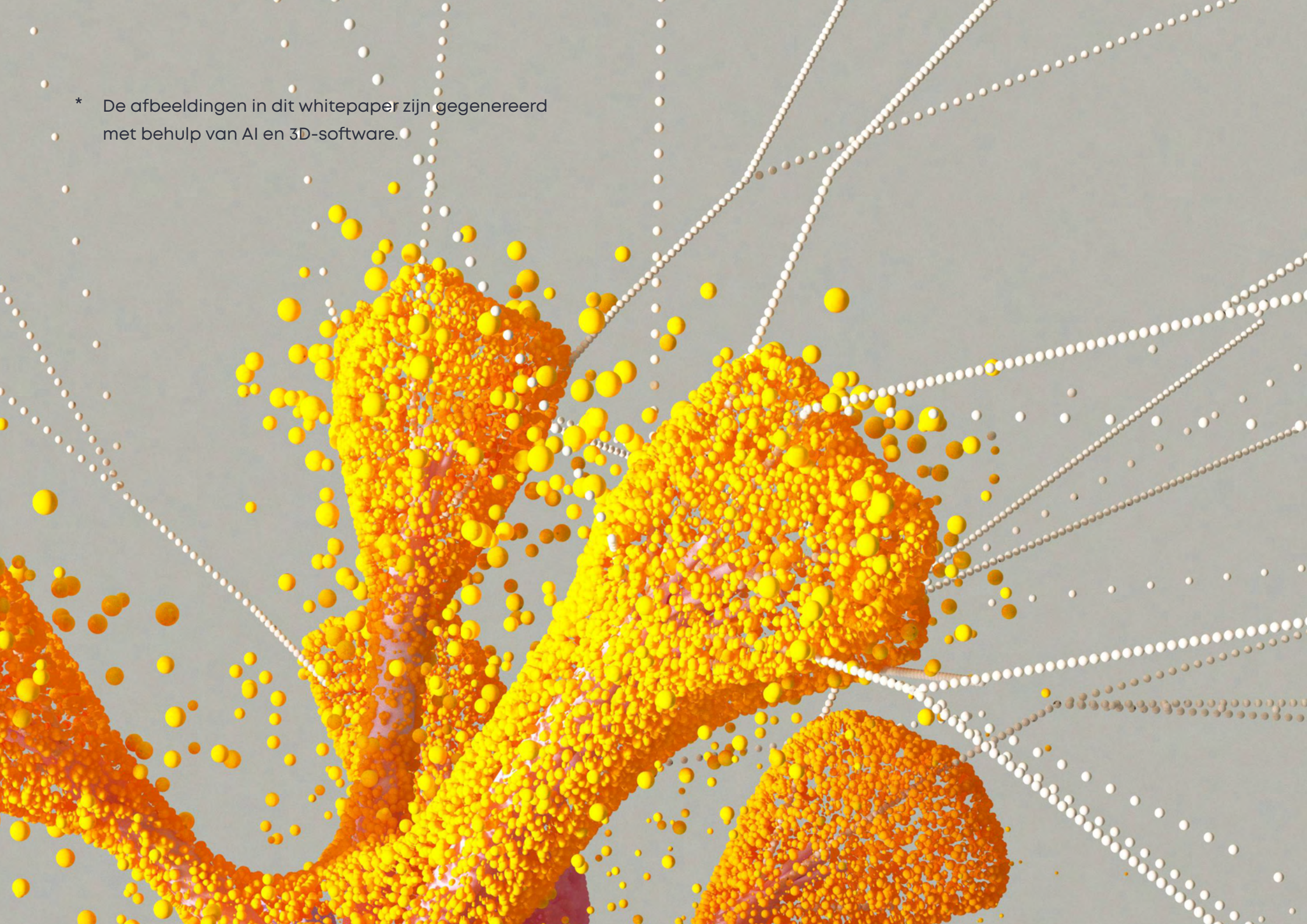


AI in software development

De toekomst van ons vak, de toekomst van jouw business

@blisdigital





* De afbeeldingen in dit whitepaper zijn gegenereerd met behulp van AI en 3D-software.

Inhoudsopgave

- 4 Waarom wij dit hebben geschreven voor jou...
- 5 Grote taalmodellen (LLM'S) in taal en code
- 7 AI-innovaties in requirements en ontwerp
- 8 De rol van AI bij testen en deployment
- 10 Uitdagingen en technische beperkingen
- 11 Hoe wij AI nu gebruiken
- 12 Wat dit allemaal voor jou betekent
- 13 Toekomstige verbeteringen en ontwikkelingen
- 14 Convergentie tussen mens en machine
- 15 Slotgedachte: creativiteit en kritisch denken



Waarom wij dit hebben geschreven voor jou...

In maart 2024 vertelde Christian Boer in ons webinar over AI-assisted development hoe wij software ontwikkelen met hulp van AI. In het webinar gaf Christian voorbeelden van wat er nu al kan (of liever gezegd: wat er in maart 2024 kon, want de ontwikkelingen gaan hard) en gaf hij zijn visie op hoe de toekomst eruit zal zien.

Dit whitepaper is een uitwerking van dat webinar, uiteraard met hulp van AI, aangevuld met nieuwe inzichten. Want welke rol je ook hebt – CEO, CFO, CTO, manager in business of IT, product owner of developer – development met AI gaat je wereld veranderen. Het lijkt nu allemaal nog in de kinderschoenen te staan, maar de ontwikkelingen gaan zo hard, dat je nu mee moet. Anders heb je straks heel veel in te halen. En omdat te voorkomen schreven wij dit whitepaper voor jou.

Welke rol je ook hebt – CEO, CFO, CTO, manager in business of IT, product owner of developer – development met AI gaat je wereld veranderen. Het lijkt nu allemaal nog in de kinderschoenen te staan, maar de ontwikkelingen gaan zo hard, dat je nu mee moet. Anders heb je straks heel veel in te halen. In het webinar gaf Christian voorbeelden van wat er nu al kan (of liever gezegd: wat er in maart 2024 kon, want de ontwikkelingen gaan hard) en gaf zijn visie op hoe de toekomst eruit zal zien. Deze paper is een uitwerking van dat webinar, uiteraard met hulp van AI, aangevuld met nieuwe inzichten.

Grote taalmodellen in taal en code

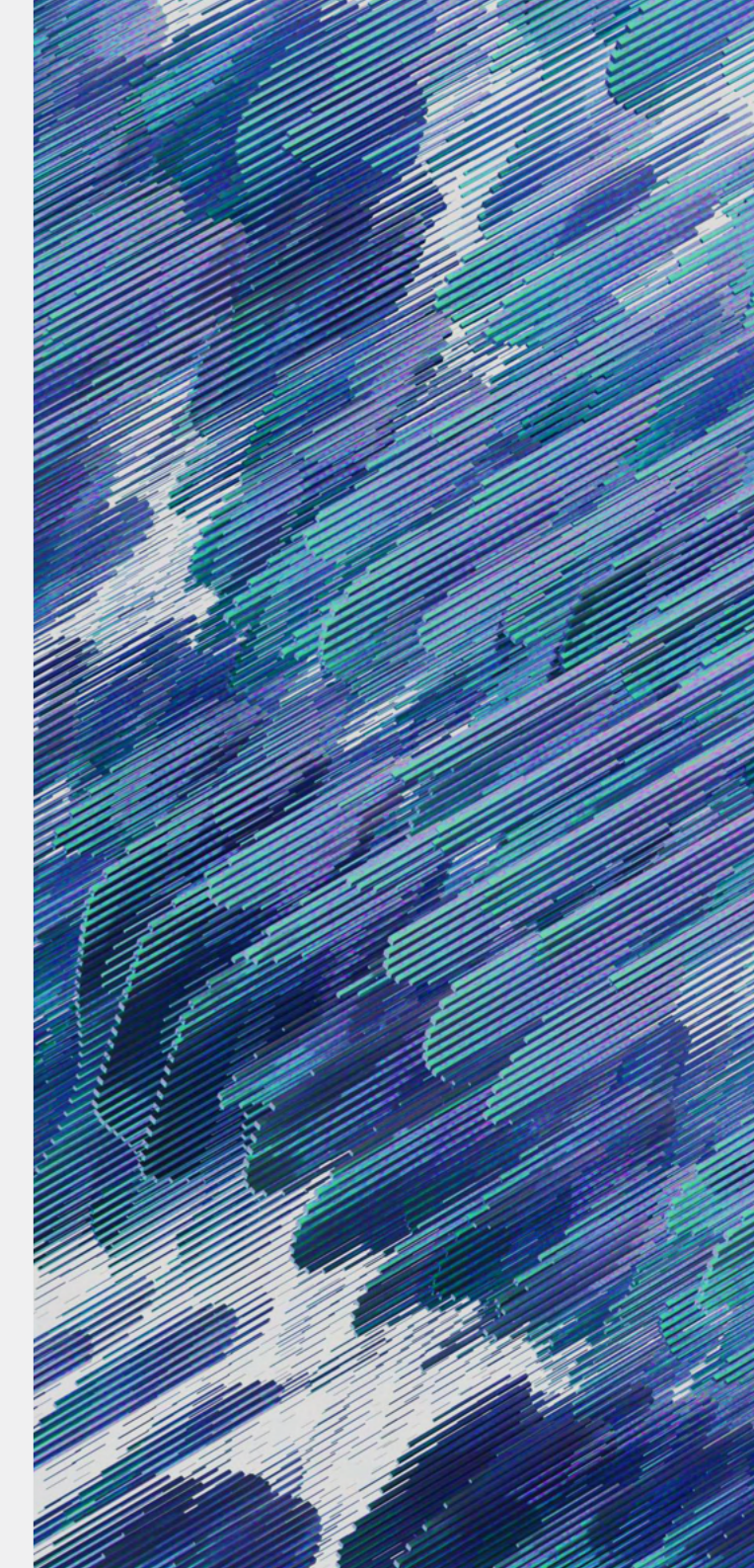
De meeste generatieve AI-tools gebruiken een groot taalmodel, large language model of LLM als basis. Voor ChatGPT, de meest bekende AI-tool, zijn dat de GPT-modellen waarvan GPT 4o de nieuwste is.

De GPT-modellen trokken bij hun introductie veel aandacht door hun vermogen om coherente en contextueel relevante antwoorden te genereren op basis van ingevoerde tekst. Toen iedereen deze kracht, door toevoeging van de chat-interface van ChatGPT, kon gebruiken was de hype compleet. Maar developers lagen wat voor op de rest van de wereld.

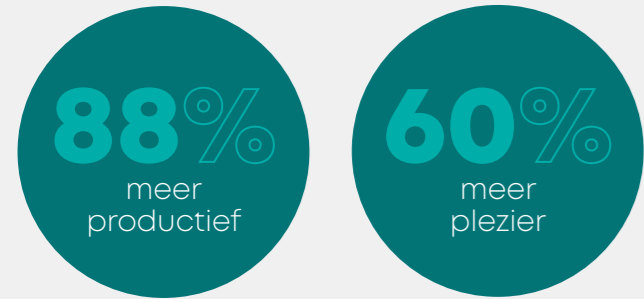
Wij hadden namelijk GitHub Copilot al, een tool die softwareontwikkelaars assisteert bij het schrijven van code. Ook GitHub Copilot maakt gebruik van een LLM, maar dan een die getraind is op de enorme hoeveelheid code die op GitHub staat. De tool werkt als een 'autocomplete': hij vult automatisch je code aan als je begint te typen. Het model voorspelt en genereert daarbij wat het volgende deel van de code zou moeten zijn.

Bliis

WORLD

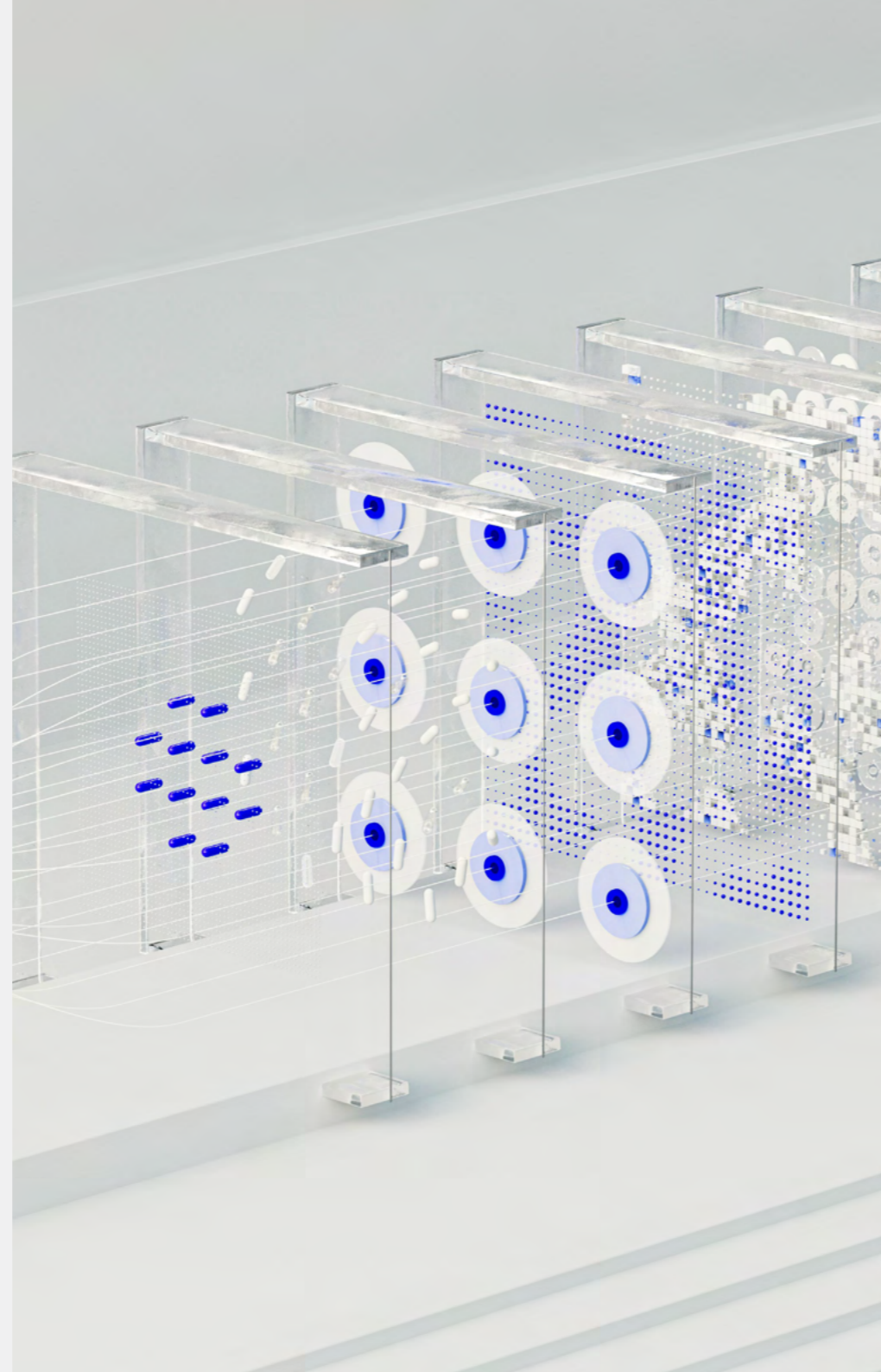


Een grootschalige enquête liet zien dat 88% van de GitHub Copilot-gebruikers zich productiever voelt dankzij de tool. De tool is zo makkelijk te gebruiken, dat zelfs niet-designers, zoals product owners, designs kunnen aanpassen. De reden daarvoor is dat je als developer niet meer uit je flow gehaald wordt door de repetitieve, mechanische taken die vaak bij programmeren horen.



GitHub Copilot is trouwens lang niet de enige AI-tool die helpt bij het coderen. JetBrains, een bedrijf dat programmeeromgevingen ontwikkelt, heeft de JetBrains AI Assistant, die developers onder andere helpt bij het schrijven van code, het grootschalig aanbrengen van wijzigingen (refactoren), het maken van documentatie en het vinden van fouten. In de ontwikkelomgeving van Amazon Web Services (AWS) kun je daarvoor Amazon Q Developer gebruiken.

Een hele andere insteek heeft Phind, dat een soort ChatGPT is waar programmeurs een vraag kunnen stellen als ze ergens niet uitkomen ('Ask mode'), of waar je mee kunt chatten terwijl je aan je code werkt ('Code mode').



AI-innovaties in requirements en ontwerp

Maar ook in de rest van het ontwikkelproces, van de eerste opzet tot het testen, zien we dat AI-modellen een steeds grotere rol spelen. Ze bieden waardevolle ondersteuning in elke fase van de softwarelevenscyclus. Ook bij het schrijven en verfijnen van user stories, een belangrijke stap in het vastleggen van de requirements van een project, gebruiken we inmiddels AI-tools.

Een mooie innovatie is bijvoorbeeld Builder.io Visual Copilot. Deze tool brengt AI het visueel ontwerpproces binnen. Maar de tool is zo makkelijk te gebruiken, dat zelfs niet-designers, zoals product owners, designs kunnen aanpassen. Zonder diepgaande kennis van designprincipes. Die heeft de tool immers geleerd door honderdduizenden designs te 'bekijken'. Als gebruiker geef je simpelweg je instructie aan Visual Copilot, zoals 'verplaats de banner naar rechts' of 'voeg een knop toe', en de tool creëert een visueel ontwerp dat past bij de rest van het project.

Niets van wat dit soort tools produceert is perfect, maar het is vaak wel een sterke basis waarop een professionele ontwerper de finishing touches kan aanbrengen. Deze vorm van generatieve AI helpt ook om mensen bij het ontwerp te betrekken die niet vertrouwd zijn met geavanceerde ontwerptools, maar wel hun kennis, ervaring en visie willen omzetten in kwalitatief goede ontwerpen.



De rol van AI bij testen en deployment

Een gebied waar AI al heel veel waarde toevoegt, is bij het schrijven van tests. AI-tools voor testontwikkeling zijn zo goed, dat er geen overtuigende redenen meer om veel handmatige tests te schrijven. Minimaal 80% van je unit tests kun je automatisch laten genereren en omdat het maken van tests (wij doen dat met GitHub Copilot) zo veel minder werk vraagt, is onze test coverage veel hoger en testen we ook 'edge cases' waar we anders geen tijd voor hadden.

Voor end-to-end UI tests (tests waarin we de daadwerkelijke interactie van een gebruiker met de software nabootsen) gebruiken we Microsoft Playwright. Het bouwen en onderhouden van dat soort UI tests was altijd heel tijdrovend, maar dat gaat nu sneller dankzij ZeroStep. Dat maakt de teststappen op basis van prompts in natuurlijke taal. We kunnen bijvoorbeeld volstaan met de instructie 'vul het formulier met toepasselijke data', in plaats van alle velden te benoemen en afzonderlijk te vullen.

Iedereen bij Blis Digital, dus niet alleen de developers, gebruikt Microsoft Copilot en/of Blis Chatty (onze GPT-4o Teams chatbot, die in onze Azure-omgeving draait) als hulp bij dagelijkse werkzaamheden. Voor testers is dat bijvoorbeeld het uitschrijven van de reproductiestappen in een bug report. In plaats van die helemaal uit te typen, genereren we die nu grotendeels op basis van de screenshots of de opgenomen video.

Kijken we naar het deploymentproces, dan zien we dat daar langzamerhand ook wat AI-ondersteuning begint te verschijnen. Microsoft Azure Copilot, de deployment-AI van Microsoft, staat nog in de kinderschoenen en is nog niet breed beschikbaar, maar er wordt hard aan gebouwd en ik verwacht dat dit ergens in de komende jaren gelanceerd wordt.



Blis

Digital

Uitdagingen en technische beperkingen

Zoals gezegd zijn de AI-tools voor developers nog lang niet perfect. Voor GitHub Copilot geldt bijvoorbeeld dat de resultaten indrukwekkend zijn bij eenvoudige opdrachten, maar dat de kwaliteit snel afneemt als de taken complexer worden.

Een praktisch voorbeeld hiervan is het coderen van een layout met een header, een footer, en een scrollbaar middenstuk. Dit lijkt misschien een eenvoudige opdracht, en ook een junior developer kan dit voor elkaar krijgen, maar met dit soort instructies stuit je al gauw op de beperkingen van het model.

Ook kan de tool nog niet met meerdere bestanden werken, waardoor veel programmeertaken voorlopig handwerk blijven. Dit is trouwens geen beperking van het taalmodel, maar een gevolg van hoe het is geïntegreerd in ontwikkelomgevingen als Visual Studio.

Hoe wij nu AI gebruiken

Binnen Blis werkt iedere developer met GitHub Copilot en JetBrains AI. Wat daarbij vooral opvalt, is hoe snel we deze tools hebben opgenomen in ons proces. Sommigen van ons hebben de tijd nog meegemaakt dat de tekst-editor als programmeergereedschap vervangen werd door de IDE (Integrated Development Environment – geïntegreerde ontwikkelomgeving).

Een ontwikkeling die niet per se onze manier van werken revolutionair veranderde, maar die ons wel hielp met allerlei kleine, terugkerende taakjes en zo veel efficiënter maakte. Dat is ook hoe het nu met AI gaat: je hebt niet door hoe vaak je het gebruikt, totdat het een keer niet werkt. We gebruiken AI de hele dag door voor kleine dingetjes en het ontwikkelen van code wordt steeds meer een conversatie met de AI. Als developer houd je je steeds meer bezig met het scherp krijgen van de requirements. De AI kan die vervolgens in code omzetten, al zul je die code nooit kritiekloos overnemen.

Blis

Uitdagingen



Wat dit allemaal voor jou betekent

Het enthousiasme over AI is groot, niet alleen in softwareontwikkeling. Zo groot, dat je het best een hype kan noemen, met als nadeel dat de verwachtingen soms ook onrealistisch zijn. Hebben we straks binnen 2 minuten, met één druk op de knop, een werkende software-oplossing voor al onze uitdagingen? Nee, zo zal het niet gaan. Op dit moment zie ik zelfs geen grote veranderingen in de doorlooptijd van softwareprojecten. Omdat het allerbelangrijkste en meest tijdrovende deel van development, precies helder krijgen wat de klant nodig heeft, mensenwerk blijft.

Berichten zoals deze van Andy Jassy, bestuursvoorzitter van Amazon, waarin hij vertelt dat Amazon Q het bedrijf 4.500 jaar aan ontwikkel-uren bespaart bij het upgraden van applicaties naar een nieuw framework, hebben dan ook wat context nodig. Het is geweldig dat AI op deze manier bijdraagt aan de kwaliteit

van software en developers tijd geeft om aan klantwaarde te werken, maar het gaat hier om relatief simpel, herhalend werk en niet om het ontwikkelen van nieuwe features. Werk dat ook geen directe businesswaarde toevoegt en dat nu daarom bij veel bedrijven blijft liggen tot de situatie onhoudbaar wordt.

Uiteindelijk gaan we natuurlijk ook op projectniveau de tijdswinst van AI zien, maar daar zijn we nog niet. Vooral omdat het daadwerkelijke coderen lang niet het grootste deel van het project is. Onze belangrijkste taak is om goed in beeld te krijgen wat jij, onze klant, nodig hebt. Wat we al wel hebben dankzij AI is de mogelijkheid om meer te experimenteren. Als een AI-tool zorgt dat je binnen een week een prototype kunt bouwen, in plaats van de 3 weken die het daarvoor kostte, dan kun je dus sneller ideeën verifiëren en besteed je minder tijd aan het maken van dingen die gebruikers niet willen.

Toekomstige verbeteringen en ontwikkelingen

Het zou sowieso een denkfout zijn om AI-technologie alleen te beoordelen op de praktische waarde van dit moment. Het is namelijk overduidelijk dat het einde van de AI-ontwikkelingen nog lang niet in zicht is. Recent introduceerde Google het Gemini 1.5-model, dat in aanzienlijk grotere contexten kan begrijpen, tot 1 miljoen tokens (stukjes tekst, zoals woorden of symbolen, die worden gebruikt als basiseenheid voor verwerking in het model).

Dit is een grote vooruitgang ten opzichte van de huidige limieten en het geeft ons een blik op wat in de nabije toekomst mogelijk zal zijn. Ik verwacht dan ook dat veel van de beperkingen van AI-tools snel zullen verdwijnen. En dat er dus een revolutie aankomt in de manier waarop ontwikkelaars omgaan met complexe programmeeropdrachten en het ontwerpen van software.

Door AI slim in hun workflow te integreren, zouden developers hun werk zomaar een factor 100 kunnen versnellen. Niet van de ene dag op de andere, maar als je naar de huidige trend kijkt zou het wel eens korter kunnen duren dan wij ons nu voor kunnen stellen.



Convergentie tussen mens en machine

Maar niet alleen de productiviteit zal stijgen. Ook de kwaliteit van AI-gegenereerde code zal nog veel verder verbeteren. De ontwikkelaars van AI-gestuurde ontwikkeltools zijn hard aan het werk om die tools de kwaliteitsstandaarden te leren die wij als developers hanteren.

De modellen kunnen inmiddels zelf, op basis van trainingsdata, best practices afleiden en die toepassen op de code die ze genereren. Dat zal over de hele linie leiden tot een convergentie tussen de kwaliteit van door mensen geschreven code en de code die uit AI-modellen komt, waarbij de twee uiteindelijk niet meer van elkaar te onderscheiden zullen zijn.

Hierdoor wordt software betrouwbaarder, omdat ontwikkelaars steeds meer kunnen vertrouwen op de consistente kwaliteit die AI-modellen genereren. Tenminste, in gangbare toepassingen. In meer gespecialiseerde software, zoals die in de medische sector waar absolute nauwkeurigheid vereist is, zullen we nieuwe uitdagingen zien. Omdat wat werkt voor een consumenten-app nu eenmaal niet per se werkt als er mensenlevens op het spel staan.

Slotgedachte: creativiteit en kritisch denken

We staan, kortom, nog maar aan het begin van een complete transformatie van hoe we software maken. En dus ook van de rol van de developer. Maar terwijl AI zich razendsnel blijft ontwikkelen, moeten we niet vergeten dat een developer veel meer doet dan code schrijven of software testen.

Want we bouwen uiteindelijk software om problemen van echte mensen op te lossen en om te zorgen dat bedrijven kunnen draaien. Dat vraagt creatief en kritisch denken en het vermogen om business-uitdagingen te begrijpen en om te zetten in oplossingen. En hoe complexer en hoe meer mission critical software is, hoe belangrijker deze vaardigheden worden.

Lees je dit en ben je CEO, business owner of manager? Dan zou dit dus mijn belangrijkste advies aan jou zijn: zoek developers en development-partners die snappen dat code maar een heel klein deel van het verhaal is. Werk met developers die echt geïnteresseerd zijn in jouw wensen en ambities en die de creativiteit en de drive hebben om die, met gebruik van alle beschikbare gereedschappen, om te zetten in software die werkt voor jou.

Verder praten over hoe jouw developmentproces efficiënter en beter kan? Daar sta ik altijd voor open:

Christian Boer



010 479 3819



C.deboer@blisdigital.com

Blis

Digital

Blis
digital

Meer weten? Check onze
website blisdigital.com of
volg ons @blisdigital



Unleash
the power of
technology