

Wat

gaan mijn

kinderen

later doen?

ten have

LEVEN EN WERKEN  
IN HET AI-TIJDPERK

Daniel Susskind

# Inhoud

Inleiding: 11

- 1 Vergeet toekomstbestendige vaardigheden 29
- 2 Verwaarloos de basisvaardigheden niet 47
- 3 Onderwijs beide, toets beide 61
- 4 Leer je te concentreren 85
- 5 Probeer niet te concurreren met AI 99
- 6 Onthoud: niet alle schermen zijn slecht 117
- 7 Laten we volwassenen niet in de steek laten 137
- 8 Word een domeinexpert 149
- 9 Sta open voor een nieuwe identiteit 165
- 10 Ga de nieuwe wetenschappelijke  
revolutie tegemoet 179
- 11 Neem AGI serieus 193
- 12 Richt je op wie je bent, niet op wat je doet 213

Conclusie 231

Dankwoord 237

Noten 239

# 1

## Vergeet toekomstbestendige vaardigheden

*We kunnen er niet van uitgaan dat er vaardigheden zijn die buiten bereik liggen – we weten gewoon te weinig over AI*

In 2014 gebeurde er iets radicaals in Engeland. De toenmalige premier David Cameron besloot dat mijn land het eerste ter wereld zou zijn waar alle kinderen, vanaf hun vijfde jaar, verplicht moesten leren programmeren. In door de overheid gefinancierde scholen door het hele land werd het oude ICT (informatie- en communicatietechnologie)-curriculum, dat kinderen had geleerd hoe ze tekst moesten opmaken in Microsoft Word en berekeningen moesten uitvoeren in Excel, afgeschaft. Daarvoor in de plaats kwam een heel nieuw curriculum, opgesteld in samenwerking met gerenommeerde beroepsorganisaties en grote technologiebedrijven, dat, in de woorden van de toenmalige onderwijsminister, ‘ieder kind zou uitrusten met de computervaardigheden die nodig zijn om te slagen in de 21e eeuw’.<sup>1</sup>

De invoering van dit nieuwe curriculum verliep moeizaam. Weinig mensen die het nieuwe materiaal op scholen moesten gaan onderwijzen bleken veel ervaring met het onderwerp te hebben of zich er prettig bij te voelen het te onderwijzen. Het lijkt erop dat Engelse beleidsmakers er in die eerste jaren van uitgingen dat, aangezien het oude curriculum en het nieuwe

allebei iets met ‘computers’ te maken hadden, degenen die het eerste hadden onderwezen met weinig moeite konden overstappen op het tweede.<sup>2</sup> Dat bleek een grote misvatting. Het maakte echter niet uit, want er werden verdere hervormingen doorgevoerd. Er werden royale beurzen ter beschikking gesteld om nieuwe leraren aan te trekken, er werden netwerken van ‘hoofdleraren’ opgezet om ervoor te zorgen dat best practices werden gedeeld en er werd een nationaal centrum voor informaticaonderwijs opgericht om de volgende generatie leraren op te leiden.

Wat in Engeland begon, verspreidde zich al snel over de wereld. Politieke leiders in steeds meer landen, die de steeds snellere technologische ontwikkelingen onder ogen zagen, besloten het Engelse voorbeeld te volgen. Sommige beleidsmakers voegden programmeren in bestaande curricula in, andere bezuinigden flink om er ruimte voor te maken. (Finland, dat vandaag de dag wordt geprezen om zijn onderwijsniveau, schrapte in 2016 staartdelingen uit zijn nationale curriculum en verving het door programmeren.<sup>3</sup>) En na een paar jaar was het moeilijk om nog een ontwikkeld land te vinden dat ‘leren programmeren’ níét op de een of andere manier had opgenomen in zijn strategie om jongeren voor te bereiden op de toekomst.

Naarmate de tijd verstreek, groeide het enthousiasme voor programmeren. Het was niet langer slechts een manier om jongeren voor te bereiden op de toekomst van werk, maar ook een manier om mensen die al aan het werk waren – misschien in functies die bijzonder kwetsbaar waren voor automatisering – te helpen overstappen naar veiliger banen. Op een bijeenkomst in New Hampshire in 2019 legde president Biden aan het publiek uit dat president Obama hem had gevraagd te bedenken wat ‘de banen van de toekomst’ zouden

worden. Het was tijd, had hij geconcludeerd, dat ook mijnwerkers zouden gaan programmeren. ‘Wie honderd meter tot een kilometer diep een mijn in kan gaan, kan toch zeker ook programmeren... en wie kolen in een oven kan gooien, kan ook leren programmeren!’<sup>4</sup>

## Een gebroken belofte

Als je terugkijkt op dat moment, begrijp je welke belofte het inhield om te kunnen programmeren. In 2007 had Apple de iPhone uitgebracht en begon het smartphonetijdperk. Binnen een paar jaar hadden we allemaal honderdduizend keer meer rekenkracht in onze zak dan NASA had gebruikt om de mens naar de maan te sturen.<sup>5</sup> Toen, in 2010, werd het AI-bedrijf DeepMind opgericht, met het onvoorstelbaar ambitieuze doel ‘intelligentie op te lossen’. Binnen een paar jaar kregen we een eerste glimp van AI-systemen die taken op zich namen waarvan we dachten dat die ruimschoots buiten hun bereik lagen. Plotseling werden onze levens doordrenkt met machines die enorm veel krachtiger waren dan ooit tevoren. En dus leek het uiteraard zinvol dat mensen zouden leren die te programmeren. Als er één baan was waarvoor we mensen zouden moeten voorbereiden, was het toch zeker die?

In 2019 kwamen de eerste waarschuwingssignalen. Dat jaar bracht OpenAI het AI-systeem GPT-2 uit. Dit was vier jaar vóór ChatGPT, het vakgebied van ‘generatieve AI’ was nog niet zo bekend, en dit was een van de eerste modellen in zijn soort. Maar wat GPT-2 intrigerend maakte voor de toeschouwers, was hoe het was ‘getraind’. Bij OpenAI wilden ze dat hun systemen tekst begrepen en op hun beurt zelf tekst genereerden. Maar om dat te doen moesten deze systemen leren hoe ze dat

konden doen door enorme hoeveelheden materiaal te verwerken: ze moesten worden ‘getraind’. GPT-1, hun eerste systeem, was getraind op een database van zeventuizend boeken. GPT-2 was echter getraind op een database van negen miljoen webpagina’s; dat was niet alleen veel meer materiaal, het was ook veel rijker materiaal, van blogs, wiki’s en nieuwsartikelen tot online fora. Zoals OpenAI had gehoopt, was GPT-2 goed in het genereren van tekst. Het veroorzaakte destijds ook enige opschudding in de pers, een voorbode van de opwindende die zou komen: ‘AI-generator van neptekst is mogelijk te gevaarlijk om uit te brengen’, schreef een krant, omdat we het verschil met het menselijke alternatief niet konden zien.

Maar toen gebeurde er iets vreemds. Hoewel GPT-2 nooit was ontworpen om code te schrijven, bleek het er vrij goed in te zijn. Opgewonden begonnen mensen op blogs en sociale media voorbeelden te delen waarbij GPT-2 slimme computerprogramma’s had samengesteld. Tijdens het verwerken van negen miljoen webpagina’s bleek GPT-2 ook code te hebben verorberd van sites als GitHub en Stack Overflow – enorme online fora vol programmeurs die code deelden en bespraken –, waardoor het had geleerd hoe het moest programmeren. OpenAI, dat aanvoelde dat er iets belangrijks speelde, gaf zijn volgende model, GPT-3, dat al getraind was met miljarden webpagina’s, miljarden extra regels code om te verwerken.<sup>6</sup> Het resultaat, Codex genaamd, was op maat gemaakt voor programmeren en werd met verbaasd gejuich ontvangen. Een invloedrijke blogger verwoordde het destijds goed: de AI, zo schreef hij, was ‘griezelig goed’.

Toen, in november 2022, werd ChatGPT uitgebracht. Het werd de snelst groeiende app aller tijden wat gebruikersaantallen betreft. En waar was het vooral goed in? Programmeren. Je vertelde het in gewone taal wat je wilde en de app

schreef de code voor je. Je hoefde niets te weten over programmeren.

Mijn dochter, destijds zes jaar oud, leerde tafels op school. Elke week kwam ze thuis met een nieuwe reeks, en ik overhoorde haar erover. In het begin was ze enthousiast, maar een paar weken later begon haar aandacht te verslappen. Ik begreep haar verveling. Het was een saaie manier om te leren. Dus zei ik tegen haar: bedenk een computerspel dat je zou willen spelen om je tafels te oefenen. Ze vertelde me iets heel fantasievol, in de trant van: ‘Aan de hemel staat een reusachtige regenboog, papa, en telkens wanneer ik een goed antwoord geef, valt er een goudklompje van de regenboog, op de rug van een vliegende eenhoorn, in mijn handen.’

Ik schreef aan ChatGPT:

*Ik zit naast mijn dochter van zes, die tafels leert. Ze heeft het volgende spel bedacht om te testen hoe goed ze die kent. Kun jij de code genereren waarmee dit spel voor haar wordt gemaakt – en me dan vertellen hoe ik de code op mijn Mac kan uitvoeren? Maak het alsjeblieft eenvoudig voor me: ga ervan uit dat ik helemaal niets van programmeren weet.*

Ik drukte op enter, en binnen een halve minuut had het de code voor me gegenereerd, met een stap-voor-staphandleiding voor hoe ik de code kon uitvoeren op mijn laptop. Ik volgde de instructie en voerde de code uit – en de eerste keer werkte het niet. Ik stuurde ChatGPT de foutmelding. ChatGPT verontschuldigde zich en genereerde nieuwe code. Die voerde ik weer uit, en dit keer werkte het. Er verscheen een venster op mijn scherm: ‘Welkom Dappere Wiskundige Ontdekkingsreiziger, Typ Alsjeblieft Je Naam’. Een knipperend vakje nodigde mijn dochter uit om te typen. Ze typte haar naam in –

en toen was ze los. Vanaf dat moment gebruikten we ChatGPT om elke paar weken een nieuw spel te genereren. (Elke versie bevatte regenbogen, goudstukken en magische eenhoorns.) Het was niet alleen leuk om deze programma's samen te bedenken, ik had ook het idee dat haar rekenvaardigheid er beter van werd.

Het was een vreemd gevoel. Ik had geen serieuze ervaring in dit soort programmeerwerk, en ik zou maanden en nog eens maanden hebben moeten leren om zulke programma's te maken (als ik onderweg het bijltje er al niet bij had neergegooid). Maar ChatGPT deed het binnen enkele seconden, naar aanleiding van een eenvoudige prompt in gewone taal, en onder het waakzame oog van een ongeduldig meisje van zes. En dit gaat niet alleen over ChatGPT. AI's van andere bedrijven bleken ook verbluffend goed te kunnen programmeren. In maart 2025 voorspelde oprichter Dario Amodei van een rivaal van OpenAI, Anthropic, dat we ons op basis van de gemaakte vooruitgang binnen 'drie tot zes maanden' in een wereld konden bevinden 'waar AI negentig procent van de code schrijft' en dan binnen 'twaalf maanden in een wereld waarin AI vrijwel álle code schrijft'.<sup>7</sup> Destijds klonk dat wellicht als sciencefiction, maar in januari meldde Anthropic dat negentig procent van de code voor Claude Code – hun AI-gestuurde programmeerassistent – door Claude Code zelf was geschreven.<sup>8</sup> Binnen tien jaar bleek de manier van programmeren die we jongeren hadden geleerd nutteloos.



## De dwaasheid van toekomstbestendige vaardigheden

Zoals het verhaal over programmeren aantoont, zijn er geen ‘toekomstbestendige’ vaardigheden. We weten gewoon niet genoeg over de relevante vaardigheden van de toekomst om jongeren te vertellen dat ze hun leven moeten omgooien voor het najagen ervan.

Het was verontrustend om erachter te komen dat we zo weinig wisten over wat AI in zijn mars had. Deze systemen, gemaakt met een bepaald doel in gedachten, bleken uitstekend te zijn in iets heel anders. Maar voor politici en beleidsmakers was het feit dat dit ‘iets’ programmeren bleek te zijn eerlijk gezegd gênant. Een centraal onderdeel van hun strategieën als reactie op technologische vooruitgang, mensen leren programmeren, was gebaseerd op de aanname dat programmeren nog enige tijd buiten het bereik van automatisering zou blijven. Dat bleek echter volledig onjuist – het was niet alleen al binnen bereik, het was zelfs juist waar AI het best in was. Is AI even goed in programmeren als de beste menselijke programmeurs? Nee, niet altijd. Maakt AI fouten waardoor een mens moet ingrijpen om de code te repareren? Ja. Maar tegelijkertijd is het nu duidelijk dat het programmeren dat op scholen werd onderwezen niet ‘ieder kind uitrust met de computervaardigheden die nodig zijn om te slagen in de 21e eeuw’, zoals de Britse minister van Onderwijs in 2015 had gehoopt. In werkelijkheid had het kinderen uitgerust om hoogstens een paar jaar te floreren.

Voor politici en beleidsmakers is het verleidelijk om deze vergissing af te doen als een ongelukkige blunder, maar dit was niet simpelweg een kwestie van pech: ze hadden het mis, en de reden waarom is belangrijk. Bovendien kunnen we,

door hun vergissing te begrijpen, misschien de waardevolste les leren die er is over hoe we jongeren moeten voorbereiden op de toekomst van werk.

Deze leiders hadden het mis omdat ze geloofden in ‘toekomstbestendige’ vaardigheden. Dat is het idee dat er een bepaalde reeks waardevolle vaardigheden is die AI nog een tijdje niet zal kunnen uitvoeren, als het dat ooit al zou kunnen – en, het belangrijkste, dat we erachter kunnen komen wat die vaardigheden zullen zijn, als we er maar zorgvuldig genoeg over nadenken. En het zijn niet alleen beleidsmakers die op deze manier denken als ze hun arbeidsmarktinterventies bedenken. Het idee van ‘toekomstbestendige’ vaardigheden is veel wijder verbreid. Loopbaanbegeleiders vertrouwen erop wanneer ze jongeren vertellen wat ze op school moeten leren als ze een goede baan willen bemachtigen. Managers doen er een beroep op wanneer ze de leer- en ontwikkelingsstrategieën plannen die hun werknemers moeten volgen. Onderzoekers beroepen zich erop wanneer ze beweren dat baan X gevaar loopt geautomatiseerd te worden, maar baan Y buiten bereik van AI ligt. Ouders beroepen zich erop bij de vraag wat hun kinderen zouden moeten doen. Journalisten gebruiken het bij het samenstellen van lijsten met ‘toekomstbestendige’ beroepen voor hun lezers.

Het idee van toekomstbestendige vaardigheden is geruststellend. De zekerheid ervan lijkt betrouwbare fundamenten te bieden waarop je een toekomstige loopbaan kunt bouwen. Maar die zekerheid is een illusie. Het geval van leren programmeren laat zien hoe dwaas het is om iets anders te beweren. Maar onze meer algemene staat van dienst is al even slecht.

Hoe onmogelijk het was om toekomstbestendige vaardigheden te identificeren zag ik pas in door de problemen van

mijn eigen vakgebied, de economie, in de afgelopen tien jaar. Wat hebben een medische diagnose stellen, een auto besturen en een vogel identificeren op het eerste gezicht gemeen? Het zijn alle drie taken waarvan vooraanstaande economen die de impact van technologie op werk bestudeerden tot voor kort dachten dat ze niet makkelijk konden worden geautomatiseerd. Maar vandaag de dag kunnen ze stuk voor stuk door AI worden uitgevoerd.<sup>9</sup> Talloze systemen kunnen medische diagnoses stellen. Vrijwel alle grote autofabrikanten hebben programma's voor zelfrijdende auto's. En het Cornell Laboratory of Ornithology heeft zelfs een app ontwikkeld die je aan de hand van een foto kan vertellen met welke vogel je te maken hebt. De allerbeste economen die probeerden te begrijpen hoe de toekomst van werk eruit zou kunnen zien probeerden keer op keer de grenzen vast te stellen tussen wat machines wel en niet konden – om er vervolgens achter te komen dat de technologische vooruitgang weinig respect had voor die grenzen. Als de toonaangevende experts die zich buigen over de impact van technologie op werk zich al zo vergisten over wat machines wel en niet zouden kunnen, wat voor kans hadden wij als leken dan?

Het andere wat me echt bijbleef, was een thema dat ik opmerkte in mijn gesprekken met ondernemers. In 2015 publiceerde ik *The Future of the Professions*, samen met mijn vader, Richard Susskind, een expert op het gebied van recht en technologie. Tijdens het onderzoek voor dat boek en de debatten en discussies nadat het was verschenen, gebeurde het talloze keren dat een professional naar me toe kwam en zei: 'Luister, wat je bespreekt is heel interessant, maar een machine zal nooit creatief zijn, en creativiteit vormt de kern van wat ik doe.' En het bleef niet bij die gesprekken. Hetzelfde argument werd in het openbaar geuit door CEO's, schrijvers, kunstenaars

# HOE BEREIDEN WE DE VOLGENDE GENERATIE VOOR OP EEN TOEKOMST MET AI?

Kunstmatige intelligentie verandert razendsnel hoe we leren, werken en carrière maken. Ouders, docenten, leerlingen en professionals vragen zich af: welke vaardigheden blijven belangrijk als machines steeds meer taken kunnen overnemen? In *Wat gaan mijn kinderen later doen?* laat econoom Daniel Susskind zien wat jongeren nodig hebben om zich voor te bereiden op de wereld van morgen. Niet door AI buiten de deur te houden, maar door te leren deze technologie verstandig te gebruiken. Zo laat hij zien hoe AI ons kan helpen, zonder dat essentiële kennis en vaardigheden verloren gaan.



**Daniel Susskind** is econoom en schrijver, gespecialiseerd in de toekomst van werk en technologie. Hij schreef onder meer *Growth*, en staat internationaal bekend om zijn scherpzinnige analyse van economische en maatschappelijke veranderingen.

**th** ten have

Vertaling Dennis Keesmaat

[www.uitgeverijtenhave.nl](http://www.uitgeverijtenhave.nl)



9 789025 914400

NUR 130