

Masterclass: Claude Code voor beginners

Bron: AI Report (aireport.nl) — automatische transcriptie via Paco.

Ben jij er klaar voor, Wietse? Ik denk het wel. Dan zou ik bijna zeggen, alsof ik Matthijs van Nieuwenkerk ben. Take it away. Toppie. Even kijken hoor. Dan zien jullie nu mijn scherm. Zeker. Toppie. En dan zien jullie ook de video. Goed, ik doe eerst even een korte introductie. Wat is Cloud Code eigenlijk? Er zullen best wel veel verschillende niveaus in het publiek zitten, denk ik. Het niveau voor vandaag is echt beginner. Dus mensen die, precies zoals Alexander zei, zelf geen softwareontwikkelaar zijn. Maar wel die behoefte voelen om zelf toffe doeltjes te maken. Daar ga ik vandaag wat over vertellen. Je kunt Cloud Code eigenlijk zien als een soort democratisering van softwareontwikkeling. Namelijk, er is een kleine club mensen die software kan maken en een hele grote groep mensen die software gebruikt. En die hele grote groep mensen die software gebruikt kan nu ook wat meer software gaan maken. En dat is best wel een verschuiving. Wat is Cloud Code? Cloud Code is drie dingen eigenlijk. Nog wel meer. Maar dit zijn de drie dingen waar ik even op wil focussen. Het is een virtuele softwareontwikkelaar. Het is een vorm van AI gedreven context engineering. Ga ik allemaal uitleggen. En een combinatie van een model en een scaffold. Ook die scaffold ga ik uitleggen. Ik onderlijn vaak even woorden. Waar je misschien van denkt. Wat bedoelt hij daarmee? Als ze onderlijn zijn. Komen ze nog terug. Je gaat daar op dubbel klikken. Zo voelt het dan een beetje, een soort linken. Om te beginnen, Cloud Code als virtuele softwareontwikkelaar. Wat is software ook alweer? Software is eigenlijk hetgeen dat draait op de hardware. We hebben allemaal apparaten en die apparaten doen heel weinig als er geen software is. Als die software ergens bij de chip komt, zijn het vooral nullen en enen. Die zien wij niet, die begrijpen wij ook vaak niet. Een hele kleine groep mensen snapt dat. En er zijn allemaal lagen overheen gekomen eigenlijk door de jaren heen van verschillende abstracties. Je kunt machinecode schrijven, dat doet ook bijna niemand meer. Assemblycode, dat is iets makkelijker, maar dat doet ook bijna niemand meer. Dan heb je de high-level languages, dat zijn de C's, PHP's, Python's, et cetera... Dat doen nog wel heel erg veel mensen, maar ook steeds minder. En de applicatie frameworks. Dat zijn eigenlijk soort verzamelingen van, ja, een soort off the shelf. Die liggen al in de kast als het ware. Dingen die heel vaak in software zitten. Een tabelletje laten zien, zoeken, et cetera. Dat zit dan eigenlijk allemaal in die frameworks. We hebben er nu een nieuwe laag bij. Natural language. Oftewel, software maken met gewone mensen taal. We hebben eigenlijk een laag erboven nu, waardoor iedereen die bijvoorbeeld Nederlands spreekt of Engels, vaak nog net even wat beter als het om taalmodellen gaat, kan nu eigenlijk ook software maken. Kan nu deze hele

laag door en eigenlijk bij die chip komen met nullen en enen zonder te hoeven kunnen programmeren. En dat is best wel een flinke stap, want software was tot nu toe een soort eigen taaltje. Dat moest je echt leren. En nu kan je eigenlijk met normale mensentaal ook software maken. Ja. Dat heet dan vibe coding. Ik zeg het een beetje half lachend, want het is een beetje een controversieel iets. Hier zegt eigenlijk Simon Willison dat het een onverantwoordelijke manier is waarop software wordt gemaakt. En mensen eigenlijk volledig prompt gedreven, dus alleen maar op basis van hun wens, hun verzoek. Zonder te kijken wat er onder de motorkap gebeurt, software aan het maken zijn. En hij maakt deze opmerking hier eigenlijk met een soort knipoog naar mensen van... Vibe coding wordt zeker gezien door mensen die daadwerkelijke programmeurs zijn en softwareontwikkelaars als niet per se altijd een goede ontwikkeling, omdat mensen zichzelf ook overschatten als het gaat om wat je met vibe coding kan. Daar kom ik straks ook op. Maar ik zeg alvast, er wordt ook veel onderschat vanuit de kant van softwareontwikkelaars wat er inmiddels met vibe coding kan. En Fypecode platforms kennen jullie misschien al. Je hebt een aantal online platforms zoals Lovable, het Europese Lovable. Bij Lovable kan je eigenlijk op de website een droom invoeren, een verzoek. En komt daar een soort appje uit, een beetje tussen een artifact, een widget, een klein stukje software en een applicatie in. Maar vandaag gaan we het over Cloud Code hebben. Dat is echt wel een stapje verder nog dan wat in Lovable kan. Ik durf wel te zeggen dat veel mensen die in lovable dingetjes hebben geprobeerd in het verleden... het nog eens moeten proberen met Cloud Code. Heel even een klein stapje terug. Software maken vanuit de gebruiker is geen nieuw idee. Lang geleden, ik bestond toen al, maar ik bestond echt net. Toen hadden we HyperCard. Zie het als een soort PowerPoint waarin je software kon maken. eigenlijk zelf al software kon maken... als gebruiker door te slepen... drag en dropen... plaatjes aan elkaar te koppelen. En hier waren toen best wel veel Apple gebruikers... zelf kleine handige stukjes software in aan het maken. Het was eigenlijk de eerste vonk... van een soort democratisering van software... Uiteindelijk liepen heel veel mensen tegen de beperkingen aan van Hypercard... en je kon er niet tegen praten. En je moest eigenlijk toch nog wel best wel technisch zijn. Maar ik wil toch even het punt maken... het verlangen van de gebruiker om meer te worden dan een gebruiker... en ook een maker, is al heel oud. Ik zei het al, Vypcoding democratiseert softwareontwikkeling. Je zou het eigenlijk kunnen zien als... ontwikkelaars zitten aan één kant van de muur... en de gebruikers aan de andere kant. En die ontwikkelaars gooien eigenlijk naar ons toe als gebruikers. Ik zeg even dat ik dan een gebruiker ben... Dan gooien we als ontwikkelaar jullie software toe. En daar heb je het maar mee te doen. En als jullie iets willen, moet je ons heel lief vragen. Of we die feature willen toevoegen. Of die bug willen oplossen. En misschien doen we dat dan een keer. Dit is een beetje de machtsverhouding tussen de ontwikkelaar en de gebruiker al heel

lang. Wat gebeurt er nu? De doos gaat open en er wordt een trap gebouwd... en de gebruikers komen over de muur heen. En dat is best wel spannend. Het is spannend omdat die ontwikkelaars... en dat is niet alleen maar wat ze dan gatekeeping noemen... dus een groep mensen die eigenlijk zegt... blijf weg vanuit onze wereld. Dat klinkt heel cynisch. Het is ook ontwikkelaars die zich echt zorgen maken van... jongens, dit is best wel een discipline... Het heeft te maken met veiligheid, het heeft te maken met data... het heeft te maken met gebruikers laten inloggen... het heeft te maken met is die software efficiënt... kan die software ook weer geüpdate worden later, noem het allemaal maar op. Software maken is meer dan alleen maar de applicatie shippen. Daarnaast een heel proces van maintenance, et cetera. Dus ik wil toch even zeggen vanuit de ontwikkelaars hart... dat cynisme en die zorg, die is niet helemaal voor niets... Maar ik wil ook aangeven dat de gebruiker nu zoveel kracht krijgt. Vind ik in ieder geval ook heel gaaf. Want je kunt voor hele specifieke kleine gebruikersgroepen best wel leuke software gaan maken. En dat gaan we ook zien. Ik wil me dus eigenlijk vandaag focussen op de gebruiker. Als je softwareontwikkelaar bent die nu meeluistert en je hebt nog nooit Cloud Code gebruikt of niet recent. Is het nog steeds interessant voor je? Maar besef je alsjeblieft, ik zit hier echt voor die eindgebruiker om ze een beetje te laten zien. De eerste stapjes richting je eigen software maken. Dan voel je al een beetje, oké, kunnen die gebruikers dan nu alles bouwen? Nee, absoluut niet. Want er zijn een soort levels van complexiteit. Complexiteit hier weergegeven in een soort abstracte vorm. Dus zie je het als een soort netwerkje wat vanuit het buitenland... Het midden groeit en het midden is makkelijk. Dus bijvoorbeeld regels code afmaken terwijl je aan het typen bent als programmeur. Dus een soort suggestie erachter. Dat gaat echt heel goed. Dat is eigenlijk de eerste toepassing geweest van die taalmodellen. En dat kan eigenlijk al een jaar of twee best wel krachtig. Veel softwareontwikkelaars gebruiken dat ook. Maar dan kan je eigenlijk al programmeren en dan laat je je zinnen afmaken als het ware. Dan kleine widgets maken. Cloud of Entropic noemt ze artifacts in Cloud. Dus jij vraagt bijvoorbeeld om, kan je een leuk boodschappenlijstje voor me maken? Wat met emojis erbij die ik dan als pdfje kan uitprinten. Dan maakt hij eigenlijk een heel klein visueel mini programmaatje voor je. Ja, ook wel interactief, toch? Met een calculatortje kan hij maken, met slidertjes en zo. Ja, dus niet alleen maar een statische pdf inderdaad, maar ook iets. Help me nou eens te kijken naar mijn pensioenontwikkeling de komende jaren. Ik heb dit bedrag nu. Help me een beetje als ik dit zou inleggen, als ik dat zou inleggen. Dingetjes die je ook wel eens op websites ziet van banken bijvoorbeeld. Van die handige calculatietooltjes. De kosten voor je auto bij de ANWB. Noem het allemaal maar op. Dit werkt al heel erg goed. Die kleine widgets hebben we eigenlijk al voor elkaar nu, zeg ik even vanuit de... Al in softwareontwikkeling. Dan kleine projectjes doen. Dit is eigenlijk waar we nu zijn. En wat bedoel ik met een klein projectje? Dat

is... Ik wil een... boodschappenlijstje voor de hele familie. Waarbij iedereen zelf dingen kan doorgeven... vanuit het gezin. En die lijstjes moeten ook nog eens... naar elkaar geüpdate worden. En er moet een beetje slim omgegaan zijn. Als er al aardappels toegevoegd zijn... moet er staan, die zijn er al. En... Dus dan ben je echt wel voorbij een ANWB-website aan het gaan... en richting een soort multi-user-achtig dingetje. Dit kan nu. Dit kon eigenlijk een half jaar geleden een beetje... en nu kan het eigenlijk best wel goed. Dat ga ik zo laten zien. Dan hebben we de grote projecten. Ik heb veel grote softwareprojecten gemaakt. Alexander ook met Blendle. De blendels van deze wereld helemaal maken met AI nu, dat is een beetje voorbij de mogelijkheden nu. Dat heeft er ook mee te maken dat een softwareproject veel meer is dan code. Het heeft ook te maken met regelgeving, het heeft te maken met gebruikers, met betalingflows, etc. Er zijn softwareontwikkelaars nu die Cloud Code ook inzetten in dit soort grote projecten, maar dan wordt wel het handje van Cloud Code vastgehouden en heb je eigenlijk een professional plus Cloud Code. Dat is niet helemaal fair. Dan heb je het grootste ding natuurlijk. En ik ga even niet uitweiden hier. Als je vaste luisteraar bent. Ken je onze AGI discussies. Dit is eigenlijk klad, koot. Voor de hele samenleving. Die hele synthetische organisaties runt. Daar zijn we op dit moment niet. En hoe lang dat gaat duren. Dat vind ik moeilijk te zeggen. De stappen hier tussen. Zijn niet stappen van een. lineaire orde in een paar jaar. Iedere stap is exponentieel moeilijker. Sterker nog, dit laatste stapje is misschien wel ineens 10, 20 jaar verder. Goed. Wat kunnen we nu? We kunnen kleine, lokale tools maken. Waarom zeg ik daar lokaal bij? Even een korte reality check. Voordat we heel erg gaan hypen over wat er allemaal kan. Want het is wel echt heel gaaf. Even een reality check. Distributie van jouw applicatie. Dus je hebt een tof websiteje gemaakt. Je hebt een iPhone app gemaakt. Je hebt een app gemaakt voor je Android TV. Dit kan allemaal inmiddels. Dan heb je die app. En dan? Hoe krijg je die dan bij je vrienden? Hoe krijg je die dan in de App Store? Kan je dan een link delen zomaar? Waar staat het dan? Dit zit niet standaard in Cloud Code. Je kunt Cloud Code gewoon een beetje naïef vragen. Hoe krijg ik deze app nou in de store? En neem van mij aan dat hij je stap voor stap gaat uitleggen... hoe je je aanmeldt bij Apple als ontwikkelaar... hoe je Xcode moet installeren, et cetera. Maar je zal dan wel gaan voelen... als iemand die daar niet in bekend is... van hé, wacht even... Wat moet ik nu beelden in de beeldfolder van Xcode? Ik dacht dat jij alles voor me ging doen. Dus daar valt als het ware een beetje het gordijn naar beneden. En is de goocheltruc voorbij. Dus distributie is nog tricky. Maar als het gaat om bijvoorbeeld die kleine websitejes. Niet onmogelijk. Dan opslag. Opslag heeft eigenlijk te maken met wat wij noemen in de softwareontwikkeling. State. Wat bedoel ik daarmee? Je hebt een gedeeld boodschappenlijstje. Op jouw apparaat staat het boodschappenlijstje... maar die moet dan ook staan op de apparaten van je kinderen. Of in de cloud. Dus jullie allemaal communiceren

met dezelfde database... met dezelfde opslag. Dus dat de status van het boodschappenlijstje... verdeeld wordt over al die devices. Dit is een oud probleem in software ontwikkelen. Dat hebben we al lang opgelost. Maar dat betekent ineens wel dat je databases moet gaan aanmaken. Of een Airtabletje kan je ook doen. Maar je moet wel een accountje hebben ergens. En een beetje begrijpen... als ik met de cloud ga praten, wat gebeurt er dan? Ook hier kun je best wel grappig aan Cloud Code vragen. Hé, zet het eens even in de cloud. En dan gaat hij toch het een en ander voor je proberen. En als je een beetje meeleeft, kan je een end komen. Maar dan komen we op het laatste punt van de reality check. Beveiliging. Dan staat het ergens in een Airtable. Maar wie heeft daar dan toegang tot die Airtable? Is er een soort gebruikerssysteem? Mag iedereen de boodschappen aanpassen in de boodschappen app? Wat als het misgaat? Kunnen we dan nog terug? Noem het allemaal maar op. Uiteindelijk zijn deze drie thema's. Dus de app verspreiden via stores of via links. De opslag van data zodat je onderling kunt synken tussen de apps. En de uiteindelijke beveiliging zijn nog niet opgelost. Ik moet wel zeggen, er worden grote stappen gemaakt. Maar ik wil toch even gezegd hebben. Focus je nu, als je niet zoveel weet van de achterliggende techniek. Echt even op leuke lokale apps. Die je misschien deelt als HTML bestand. of lees eens hoe je TestFlight kan gebruiken... om een iPhone-appje naar een vriend of vriendin te sturen. Is het herkenbaar voor je dat dit drie hekelpunten zijn? Ja, het is een beetje doorzetten. Het vraagt doorzettingsvermogen en ongemak in jezelf proberen te onderdrukken. Want als je maar lang genoeg doorvraagt, dan helpt hij er echt wel bij. Maar ik snap wat je zegt. Ja, dat is een beetje het gekke dat Cloud Code kan code schrijven... Maar kan je ook als mens begeleiden door het aanmaken van een Xcode developer account... of een Apple developer account in Xcode? Dat kan die ook. En dan wisselt die eigenlijk van programmeur naar een soort technisch maatje die jou helpt. En dat is zo een beetje gek nog aan die AI-systemen... dat ze zo kunnen shapeshiften van rol eigenlijk midden in jouw proces. Wat is Cloud Code nummer twee? Cloud Code is een vorm van AI-gedreven context engineering... Top. Contact engineering houdt eigenlijk in... een vorm van prompt engineering. Prompt engineering kennen we eigenlijk als... de kunst van de goede vragen stellen aan een AI. Er zijn hele cursussen die mensen daarin hebben gevolgd. Context engineering is eigenlijk... Hoe zorg je nou dat dat statische model. Want al die AI modellen zijn eigenlijk modellen die niet zich aanpassen terwijl je met ze praat. Die moeten zich aanpassen qua kennis met wat jij ze op dat moment geeft. Ze moeten het zelf gaan opzoeken op internet. Of ze moeten jouw pdf je lezen. Maar je moet ze eigenlijk helpen door ze zoveel mogelijk context te geven. En dat kan je op allerlei manieren doen. Dat was dat. Een vorm van grounding door middel van documenten met Copilot bijvoorbeeld. Of je kunt zeggen, je mag het web op. Of hier heb je een mail van de laatste jaren. Trek daar eens een conclusie uit.

Dan ben je eigenlijk die context een beetje aan het vullen. Net wel een stap voorbij prompten. Ik heb het even visueel gemaakt voor jullie. Het model is statisch. Die modellen komen iedere drie tot zes maanden uit. En dan heb je een model wat je eigenlijk daarna vooral moet gaan helpen... om beter te worden dan dat het eigenlijk van zichzelf is. En dat kan je doen door zoveel mogelijk en niet zoveel mogelijk. Dat is al een fout om dat zo te zeggen. door zo kwalitatief mogelijke context toe te voegen. Meer is zeker niet altijd beter. Dus dan heb ik het echt over een goede prompt... maar vooral ook over goede voorbeelden... van wat je bijvoorbeeld zou willen zien aan het einde als die klaar is. Toegang tot bepaalde documenten die belangrijk zijn. Handleidingen van APIs die je misschien wil integreren. Als je zegt, ik wil de laatste berichten van NOS ophalen of van de NS... dan moet je wel die NS-API-documentatie even meegeven... of hem aangeven, ga eens even op zoek als je wil. En dan vul je daarmee eigenlijk die context. Waarom vertel ik dit? Het is eigenlijk best wel technisch. Omdat dit iets is wat Cloud Code ook deels voor jou doet. Het is eigenlijk best wel gaaf. Cloud Code is zo krachtig, omdat het eigenlijk zelf een beetje probeert te bedenken... wat is eigenlijk allemaal handig om toe te voegen aan die context. Je kunt met hele rauwe prompts, dat ga ik je zo laten zien... eigenlijk al best wel snel gave dingen maken daardoor. Wat er vaak misgaat, heel even kort, is dat mensen veel te lange prompts gaan doen. Het context window te klein is. Mensen weten vaak niet eens hoe groot het context window is. Dit kan je opvragen binnen Cloud Code met slash context. De inhoud is heel ongestructureerd. Dus je denkt meer is beter. Ik klap er even wat documentatie in. Nee, ik doe alleen de documentatie erin. En die wezenlijk belangrijk is. Want anders krijg je een soort concurrentie binnen de context heet dat. Dus je zegt eigenlijk tien tegenstrijdige dingen. En dan moet Cloud een soort van gaan bedenken welke nou de waarheid is. Geef dan de waarheid. Doe wat vooronderzoek. En je informatie kan oud zijn. Dus je bent al heel lang aan het praten in Cloud Code. Doe soms even een refresh. Je kan een slash clean doen. Dan gooi je even alles weer weg. En dan begin je even opnieuw. Je gaat dit straks laten zien. Ja, absoluut. Nou, ik zei het al, Cloud Code doet een groot deel van de context engineering voor je. Dat doet hij eigenlijk door middel van een scaffold te zijn. Dit is het laatste technisch wat ik vertel, dan gaan we een stuk praktischer hoor. Een scaffold is eigenlijk iets wat je om een model heen zet. Jij noemt het altijd de Superman-pak. Dat vind ik een hele mooie metafoor. Waarbij je eigenlijk zegt dat model zelf krijgt allerlei tools en trucjes en gereedschappen om zich heen. Om eigenlijk de wereld in te kunnen. Om documentatie te kunnen gaan lezen. Om zijn eigen context te bepalen. Om de prompt te herschrijven die hij matig vindt. En zo helpt eigenlijk die scaffold om het model eronder een stuk krachtiger te maken. Je kunt het eigenlijk zien als een soort hekjes bij de bowling. Bij de bowling kan je een kind hekjes geven. Waardoor die bowlingbal eigenlijk, in dit geval het taalmodel klat. Een beetje tussen de hekjes blijft. Zo'n

scaffold biedt een soort hulp. Van joh, je gaat de verkeerde kant op. Stop maar weer. Of je bent wel heel erg lang aan het nadenken. Of je hebt net die website getest. Je bent al tien seconden aan het testen. Het is goed zo. Het is eigenlijk een soort Alexander bij Wietsen tijdens de podcast. Wiets, het duurt te lang. Daarover gesproken, het duurt te lang. Het moet erdoor. De scaffold. Nou, de scaffold zit dus om het model heen. Dus je hebt het model, de context en de scaffold. In ons geval staat cloud in het midden. Dus dat is het model dat jullie kennen als 4.5 Opus. Daarnaast heb je context en daarnaast heb je cloud code. Wat cloud code heel gaaf kan doen, is bestanden bewerken op jouw computer... en commando's uitvoeren, de context managen en met cloud praten. En eigenlijk is de spin in het web cloud code. Dus als je je afvraagt, wat is dan het verschil tussen cloud en cloud code... Claude Coat is de rapper, is het Superman-pak. En Claude is het onderliggende model. En die twee kennen elkaar. Want die zijn ontwikkeld door Entropic samen. En Claude is inmiddels gefintuned op zijn scaffold. Dus die weet, oh, Claude Coat, ik ken jou. Ik weet helemaal wat jij wil. En dat is zo megakrachtig aan deze combo. Dus eigenlijk is Claude in het midden veel beter geworden in de laatste zes maanden. En Claude Coat is beter geworden. En we zijn toch wel over een soort dammetje of... Crossing the Rubicon... zijn heel mooi vanochtend gegaan... wat betreft software maken. Voorbeelden, daar wilden we naartoe. Ik heb van jullie allemaal gave ideeën gehad... over software. Ik moet zeggen... sommige waren dingen als vrede in de hele wereld. Ik heb die er niet ingeplakt. Ik vond hem toch te groot als prompt. Maar wat ik heb gedaan is... ik heb er een 4 geselecteerd. En ik dacht... Ik plak ze gewoon één op één in Cloud Code. Oké, want jij hebt gewoon uitgebreide beschrijvingen van kijkers gekregen met wat ze ongeveer wilden. Ja, geen letter aangepast. Geen letter, want kijk, jij hebt mij er wel eens op gewezen. Ik heb een soort handicap nu. En daarmee bedoel ik zeg maar handicap. Als in, als softwareontwikkelaar weet ik eigenlijk te veel. Ja. Kijk, ik wil eigenlijk tegen Klaat zeggen... Jij hebt eigenlijk de gemiddelde kijker nagedaan. Ja, ik weet eigenlijk te veel. Dus ik zeg dan eigenlijk tegen Klaat, het liefst zelf... Ja, ik wil graag dat je er een Postgres database aan maakt die later... Ik ben dan heel opinionated met heel veel mening en heel veel sturing. Ik ben aan het micromanagen eigenlijk. Terwijl, ja, wat je eigenlijk moet doen is gewoon... Ik wil een app maken voor Zuss en Zat. Plak er maar in. Wat heb ik gedaan? Eerste app die ik erin heb geplakt. Ik zou een app willen bouwen die food pairing advies geeft... voor non-alcoholische dranken. Letterlijk de zin. Nou, klat ging aan de gang. Wacht, wacht, wacht. Want dit klat ging aan de gang. Dus ik laat het zien hier. Dus ik plak hem erin. Ik ga zo uitleggen waar ik dit plak. En hoe je dit installeert, maakt niet uit. Even wat je kan verwachten. Ja, precies. Dit is op versneld. Ja, dit is keer 800 procent. Keer acht. Want Opus, die op de achtergrond draait, is mega traag, vind ik. Met traag bedoel ik acht minuten later. Single shot had ik deze hele applicatie. Oké, acht minuten is hij nu aan het pruttelen.

Ja, dus aan de linkerkant zie je eigenlijk... Wietse die plakt de prompt erin. Die drukt op het oranje knopje. En aan de zijkant begint er van alles te rommelen. En mijn doel was geen letter aanpassen aan wat jullie me hebben gestuurd. En single shot. Oftewel, daarna moet het in één keer werken. Dat vind ik gewoon vet. En dat is bij deze allemaal gelukt. Het is echt gestoord. Ik ga wel een beetje uitleggen waarom het bij sommigen nog iets moeilijker was. Er wordt een heel plan gemaakt. To-do's worden afgewerkt. En dan acht minuten later hebben we dit. Dit ding. Eerst wat misschien mensen zeggen. Het ziet er zo eijerig uit met dat paars en dat blauw. Dat is weer zo'n typische AI applicatie. Dat komt omdat ik gekozen heb voor Vanilla Cloud Code. Dus geen extra trucjes. Geen extra NPC's of tools of skills. Ik wilde het zo eerlijk mogelijk maken. Wat als jij als kijker nu zelf Cloud Code installeert en die prompt er gewoon in plakt en verder geen enkel extra trucje doet. Wat krijg je dan? Wat zien we hier? Wat je hier ziet is eigenlijk, je kunt een, ik zal hem even opnieuw starten. Je kunt een gerecht selecteren en dan krijg je daarbij eigenlijk alle drankjes, non-alcoholische drankjes die goed passen bij dat gerecht. En het grappige van dit ding is, ik ben redelijk simpel begonnen. Hier zit niet eens een database achter. Hij heeft het eigenlijk gewoon helemaal statisch uitgeschreven. Je kunt wel een beetje zoeken, maar ik vind het wel grappig hoe hij dit gedaan heeft. Hij is trouwens ook mobile friendly. Dus als je hem klein maakt, kan je hem ook op je smartphone. Doet hij allemaal vanzelf erbij. En creëert eigenlijk best wel een cute ding. Hier heeft dus Klad ook de inhoud ingestopt. Ja. Hij ging op een gegeven moment ook zoeken. Hij zei, ik ga een beetje zoeken naar websites die hierover praten en blogjes lezen en zo. Want ik moet het zelf ook nog een beetje aan wennen. Maar deze AI-programmeur is natuurlijk ook een research assistant. Dus die is ook op zoek gegaan naar de inhoud van hier. Ik had ook in eerste instantie kunnen zeggen, ik ga zelf op zoek naar een interessant boek. Waar ik de rechten op heb. En ik verwerk dat boek tot een app. Maar ik heb hier echt Carplunge gezegd, zoek het maar uit. Dan veel breder. Ik las deze en toen dacht ik... oh nee, dat kan niet. Dat moeten we niet doen. Dat is natuurlijk veel te groot. Dus iemand vraagt... ik wil een roosteren app voor een mbo school... die rekening houdt met inzetbaarheid van docenten per dag... lesdagen van studenten, et cetera. Heel, als je softwareontwikkeling hebt gestudeerd... typische opdracht om te bouwen. Want je moet rekening houden met... eigenlijk collisions, dus conflicten binnen roosteren... namen die veranderen, dingen die moeten verschuiven, et cetera. Ik heb weer, as always, ik heb hem erin geplakt. Hier is hij al naar beneden geschoven als een van de sessions. Dus de session staat Build School Schedule App with Teacher Availability. Die titel geeft hij zelf aan die sessie, kladcode. En die gaat hier aan de gang. Die was dan op een gegeven moment klaar. Dan krijg je de MBO rooster app. Single shot draaide dit hele ding. Dus ik kan hier, ik ga dat ook doen. Een docent toevoegen, vakken, groepen. Uiteindelijk, als je dat dan allemaal invoert... gaat hij roosters

maken die je kunt bekijken. Voor de duidelijkheid. Kun je dit nu meteen in je mbo-school inzetten? Nee. Want hoe zit het met beveiliging? Wie mag het rooster aanpassen? Hoe krijg ik de studententoeegang? Maar werkt het? Ja, helemaal. Als in, het is... Het conflicten oplossen. Het is best wel bizar. Hier zat ik voor het eerst. Dat ik dacht... Oké, jeetje, dit is wel iets wat een half jaar geleden nog niet echt goed ging. Want ik wil toch even gezegd hebben... tekst geplakt vanuit de vraag. Gedrukt, gewacht, geopend. Ja, je hebt echt niks gedaan. Ik heb echt niks gedaan. En ik heb ook gedaan alsof ik niks wist. Ik ben later wel gaan kijken hoe hij dit allemaal doet. Uiteindelijk heeft hij dus geen database gemaakt... maar een soort bestandje waar hij alles in bijhoudt. Heel cute, zeg maar. Maar ik had hem ook kunnen vragen... joh, ik zou het wel prettig vinden als je eigenlijk gewoon een database zou gebruiken. Maar ik ben gestopt. Ik heb gedacht, dat weet ik. Wat kan je hiermee? Toen dacht ik... Hé, maar dit moet toch in de cloud? Leuk zo'n roze systeem. Ja, maar je kan niet als URL delen met een collega op dit moment. Want het draait letterlijk op jouw eigen computer. Ja, en ik heb hier heel specifiek gevraagd. Kan je zorgen dat de data opgeslagen wordt in de cloud? Want op zich kan ik het bestandje nu met jou delen. Maar als jij hem dan opent, heb je je eigen roze systeem. Ja, je zou letterlijk de HTML-faal met mij kunnen delen en naar mij mailen. En dan open ik die. Maar dan kunnen wij nog steeds niet de onderliggende database delen, want die staat op jouw computer. Ja, dus ik heb eigenlijk gewoon... Dus ik denk, hoe zal ik dit heel schattig vragen? Kan je zorgen dat de data opgeslagen wordt in de cloud? Ik had zelf allemaal ideeën over databases en zo, maar ik dacht, laat maar gaan. Hij ging weer lekker pruttelend, zei hij. Hier is die, je moet het even bij Firebase neerzetten. Dan heb je deze kosten. De compliance, et cetera. Compliance, even bij je. Goed, hij heeft het erin gezet. En toen verscheen er ineens in de hoek dit ding gesynchroniseerd. En als je hem opent op een andere computer, dan is de data daar ook. Zonder dat je inlogt. Dus iedereen die hem opent, is gesynchroniseerd. Dus ik wil nog even terugkomen op mijn reality check. Je hebt geen authenticatie. Je hebt een sharing van een database. Iedereen kan daar nu in schrijven. En dit is eigenlijk de glazen muur. De glazen barrière. Waar je nu als leek die niet zover maakt. Gewoon tegenaan gaat lopen. Wat een leuk rostersysteem. Maar kan ik het al uitrollen naar mijn leerlingen? Dan ben je nog wel drie, vier stappen verder. Wil ik toch even gezegd hebben. Maar wat hier gebeurt in één keer. Qua onderdelen. Ik ben echt wel onder de indruk. Toen zei ik, oké, ik wil een app bouwen die op mijn telefoon draait. Mijn hersenspindel kan inspreken. Ik zei dit niet, dit heeft iemand gevraagd. En ik denk, ik dacht nog, ik moet toevoegen. iPhone of zo. Ik dacht, nee, weet, afblijven. Dus ik wil afblijven. Even om te begrijpen, een hersenspindel kan inspreken, waarna die automatisch naar mijn e-mail wordt verzonden, om op deze manier later een soort to-do-app maken met spraak. Noem het een shoutbox of zo. In inbox. Meestal betreft het voor mij zogenaamde to-do's. Mensen

terugbellen, dingen uitzoeken. Ja, oké. Jouw neiging was om hier details te gaan toevoegen. Ja, omdat ik dacht, ja, e-mail verzenden. Er moet een hele e-mail fulfillment delivery service achter. Die audio moet gecompressed worden. Mijn hele nerdbrei ging natuurlijk aan. Maar toen dacht ik weer, nee, nee, niet aankomen. Nou, dit was in Cloud Code, in Visual Studio Code. Ik zal het wel even onderscheid uitleggen. Daar heb ik een mooi plaatje van gemaakt. Maar ik draaide Cloud Code hier even op een andere plek. Dat was helemaal niet nodig. Maar omdat ik dan blijkbaar dacht als ontwikkelaar dat dat hoefde helemaal niet. Nee, jij dacht dit is te ingewikkeld. Maar oké, ja, komt zo. Hier was hij zo op een gegeven moment klaar. Toen zei hij aan het einde, oh ja, hier, open Xcode. Maak een nieuw project. Doe even je storage naar Swift data aanzetten. En ik kan dit allemaal, want ik heb iFord apps gemaakt. Maar ik dacht, oké, weet je wat? Als ik hem alleen maar hoef te drag en drop in Xcode en op beeld hoef te klikken, vind ik het nog wel cool. Dus ik heb hem gedrag en drop in Xcode. Hele app. Haha. Nee, maar echt. Dus ik heb hem op mijn telefoon gezet. Hij staat nu op mijn telefoon. Ziet er ook vet uit. Ja, dus ik heb hem op mijn telefoon gezet. Dus ik druk op opnemen. Ik krijg daarna een mail van mezelf. Yo man. Ik zeg, wow. Maar weet je wat zo grappig is? Want voor de ontwikkelaars die kijken, die denken natuurlijk, hoe stuurt hij nou die mail? Ja. Ja, natuurlijk gewoon via de built-in mailclient. Superslim, want je hebt gewoon een mailservice in iOS. En hij zegt ook in zijn reasoning trace, kladcode, ja, ik kan een heel delivery ding gaan doen, maar de meeste iPhone gebruikers hebben een mail app. Dus ik dacht zelf, wat fijn, pragmatisch, wat slim. Nou, ook, nou, hij heeft allemaal hele pragmatische dingen gedaan. En ik heb deze app, die dus hersenspin heet, hij heeft klad ook allemaal bedacht. Ik gebruik hem nu best wel chill. Met dank aan de luisteraar die dit prompt heeft ingestuurd. Nou, dat... Toen kreeg ik een beetje onderaan de lijst, zag ik deze. Ik heb hem even ingekort. Een mooi verhaal, een ernstig verhaal. Een mooie toepassing, misschien moet ik het zo zeggen. En ik dacht, nou ben ik wel benieuwd, die plak ik erin. Maar ik dacht wel, ik wil hem eigenlijk aanpassen, want ik wil eigenlijk wel een iPad-app. Het gaat om geld. heb een dochter van 7 met ernstig non-verbaal autisme ik zou voor een app willen bouwen die haar dag week maand of jaar in persoonlijk aanpasbare pictogrammen zou kunnen weergeven op dit moment begrijp gebruiken bijvoorbeeld digitale pictogrammen set van thuis bij muis die pas ik soms zelf aan print ik uit laminieren ik ja het is soort van visuele structuur van Ja, ik zie het wel als mensen die een ketting dragen met aan de kaartjes. Zodat je een beetje weet hoe de dag loopt. Zo'n soort dagstructuur te hebben. Eerst ging die dus ook weer single shot in plaat koten. Dus één keer starten en daarna openen. Toen zei hij eigenlijk hier kun je op een link klikken. Ik heb nog hier wat dingen. Dit was hier. Dus ik zei oké. Dan heb ik hem geopend. Daarbinnen kreeg ik in deze webapplicatie de mogelijkheid om als ouder dingen aan te passen. Je kunt activiteiten toevoegen. Hij heeft al die activiteiten zelf bedacht.

Iconen erbij verzinnen, et cetera. Je kunt routines aanmaken, routines beheren. Het zat er allemaal in. Een compleet instellingenscherf met allemaal customizations. Ik heb niks gedaan. Maar toen dacht ik... Ik heb nu die coole hersenspin app gezien. Dit moet gewoon een native iPad app zijn. Dus toen heb ik één regel toegevoegd aan de prompt. Eerlijk toegegeven, hij moet gewoon iPad native zijn. Toen zei hij, is goed. Tarmata is een app uitgekomen met pincode instellen. En die er serieus goed uitziet weer. Ja, ik bedoel, binnen Apple gaat het vaak sowieso wel goed of zo. Omdat die defaults wat beter zijn, maar... Ja goed, ik zat hier doorheen te klikken en ik was echt even stil. Omdat ik bedoel, ik werk echt veel met deze tooling. We zitten er middenin. Ik had over de kerst al gelezen online dat mensen zeiden... jongens, ik ben van mening gaan veranderen. Dit wordt toch wel heel bizar nu. En ik snap nu een beetje waar dat vandaan komt. Want ik had het ook. Oké, nou de vraag die nu brandt is natuurlijk hoe? Kijken. Zeg het eens, Wiets. Was ik op tijd? Ja. Net op tijd, hè? Hoe kan ik dit zelf doen? Ik kijk nu. Superleuk, Wiets. Hoe kan ik dit doen? Je kunt het op vier verschillende manieren doen. Via de command line binnen Visual Studio Code, binnen de Cloud desktop app en via de browser. Hoe ziet dat eruit? Dat is deze uitdaging. Alexander en ik zitten al heel lang in de bovenste command line. Dat is een hele spannende plek voor heel veel mensen. Die komen niet zo vaak in hun DOS terminals, als ik die wel eens noem. Toen kwam Cloud Code in Visual Studio Code. Daar zitten ook bijna helemaal geen mensen in... behalve softwareontwikkelaars. Toen kwam Cloud Code eigenlijk op het web. Dat is de laatste stap. En sinds 6 januari de middenstap in de app. Ja, gewoon de cloud app zoals je die downloadt op de computer. Dus mijn advies zou zijn... Download de Cloud App voor Windows en voor Mac. Of voor Mac, aan de hand van jou. Ik heb dit allemaal op Mac gedaan. Maar Windows is de Cloud App echt niet veel anders. Het is een hele prettige middenweg. Waarom? Je bent niet op het web bezig. Dus je kan lokaal dingetjes rommelen. En dat is gewoon fijner. Hij kan ook lokaal dingetjes uitvoeren en zo. Je hoeft niet GitHub te begrijpen. En repositories en zo. Want dat vinden mensen dan toch ineens een heel technisch iets. Maar je zit ook niet in een soort spannende terminal. Ja. Hoe kan ik dat zelf beginnen? Dan moet je dus switchen. Dan heb je bovenin... Ga ik laten zien. Oké. Dus één waarschuwentje nog. Typisch Wietse. En dit is een paar keer gebeurd bij mensen. Het is niet helemaal een gekke waarschuwing. Claude draait op jouw computer. Claude kan dingen uitvoeren. Vraagt iedere keer of hij dat mag. Maar lees je dat ook iedere keer? En het kan zijn dat hij zegt, ik verwijder je documenten. Het gebeurt bijna nooit. Het is een aantal keer bij mensen gebeurd, heb ik begrepen. Dus ik wil er wel even bij zeggen. Doe alsjeblieft Cloud Code lokaal op een machine waar je goede backers van hebt. Ik wil het echt even gezegd hebben. Want je baalt wel. Het is best wel een experimentele technologie. Dan ga je naar de website van Cloud. Dan druk je daar op Cloud Code. Dan druk je op Mac als Windows of een Windows laptop van

Snapdragon ARM. Dan staat daar in mijn geval Cloud voor Mac. Druk je op Get Started. Dan log je in. Ik ga er dus vanuit dat je een Cloud Pro abonnement hebt. Mag even bijgezet worden. 20 euro per maand. 20 euro per maand. Dus dat moet je wel hebben. En dat is ook niet ongelimiteerd. Je kunt op een gegeven moment krijgen. Ik heb dat vaak gehad. Jongen, het is even klaar voor vandaag. Dan moet je opeens 90 euro per maand gaan betalen. Precies. Of geduld hebben, tot je weer mag. Dan kom je in de Cloud App. Dan ben je nog niet in Cloud Code. Dat is waar Alexander al op doelde. Er zit een knopje, hier, Code. Daar moet je even op drukken. En dan switch je naar de codingmodus. Die ziet er zo uit. Dan zie je ook dat oranje robotje. Wat eigenlijk Cloud Code geworden is als een soort grapje. Dan ben je er nog niet. Je moet je nog heel even een paar klikjes doen. Je gaat een map selecteren. Ik zou zeggen, zet een map op je desktop. Die noem je Test App of Leuke Test. En die selecteer je. Ja, en daarin gaat hij eigenlijk werken. Dank je wel. Je moet een map maken waarin dat ding de vrijheid heeft. Het minimale, dat was vroeger nog veel moeilijker. Maar het minimale is wel een mapje. Dat mapje is ook als het goed is de zandbak waar die binnen blijft. Precies. Als het goed is. Dan plak je je prompt. Zoals ik hier heb gedaan. Je ziet het ook. Ik plak letterlijk die hersenspinselprompt erin. Nog voordat ik druk op versturen doe ik nog even een kleine tweak. Zo. Even naar Opus. Want hij staat standaard op Sonnet. Dat doet Anthropic. Want hij is veel trager en duurder. Maar ik vind wel. Om het spektakel te voelen. Zoals ik dat heb gevoeld met die verschillende apps. Moet je Opus gebruiken. Want Opus is over die Uncanny Valley gegaan. Wat mij betreft. Daar klik je op. Dan druk je op het oranje knopje en dan begint eigenlijk het videotje zoals je net hebt gezien. Dat ding gaat pruttelen, dat zegt hij ook, puzzelen, putteren, al dat soort dingetjes. En uiteindelijk zegt hij dan, hier is wat je moet openen, hier is de URL. Ik raad je aan, hou het lekker bij web. Als je snel leuke resultaten wil, dan krijg je dan een link waar je op kan klikken en die zie je op je eigen computer. Ja, dan is hij letterlijk, heeft hij een link naar het bestand op je computer. Daar kun je wel op klikken, die opent in je browser, maar hij draait op je eigen computer. Absoluut, ja. En je kunt dan zeggen, ik wil dat dit draait in de cloud. Ik wil dit nu met vrienden kunnen delen, help. Hetzelfde vraag. Niet eens aangeven hoe je het moet oplossen, maar gewoon je wens delen. En dan ben je constant bewust van het feit dat je daar niet persoonlijke data gaat delen. Dat je iets hebt gebouwd wat je zelf nog niet helemaal begrijpt. En dat het allemaal heel leuk is. En ik moet zeggen, een applicatie om een dagindeling te maken voor iemand en dat soort dingen. Zeker als het lokaal op je iPad is, is fantastisch. Er zijn zoveel dingen die helemaal niet zo beveiligingsspannend zijn per se. Ik ga je even nog een paar slash commands leren. Dan denk je misschien, doe niet zo nerdy. Je gaat al heel snel merken als je met Cloud Code praat, ik wil meer. Ik wil toch nog even wat meer. Wat doe je daar eigenlijk mee? Ik wil meer. Ik merk dat hij vergeetachtig wordt. Het is irritant. Wat is er aan de hand? Dan kan je

bijvoorbeeld doen slash context. Als jij bedoelt hij vergeetachtig wordt... dan betekent het dus... je hebt eenmaal een eerste versie gemaakt van wat je wil. En dan besluit jij... we doen het nu met één prompt... maar je kan ook zeggen... oké, maar nu wil ik dat het logo rechtsboven komt te staan. Of oké, maar nu wil ik dat er een settingscherm bij komt... waarin ik dingen kan aanpassen... Je gaat, als het goed is, itereren. Je gaat itereren door te praten met dat ding in stapjes. En jij zegt, als je dat doet, dan ga je erachter komen... dat je soms tegen beperkingen aanloopt. Absoluut. En ik denk dat... Ik moet zeggen, ik heb getwijfeld om dit uit te leggen. Omdat ik dacht... Misschien hoeft het al niet meer over drie maanden. Echt waar. Maar goed. Maar goed, nu wel. En we gaan daar tegenaan lopen. Ja, en ik wil mensen toch een beetje helpen. En ik heb een beetje verteld over context engineering en die hamburger laten zien. En onder water is Cloud Code van alles aan het doen qua magie. En dan kan jij hier meekijken. Dus als jij slash context doet, dan krijg je... Het ziet er heel nerdy uit, maar dan krijg je eigenlijk... een overzichtje van wat zit er nu in de context. Zegt hij, nou, ik heb wat system tools erin zitten. Je prompt zit erin. En zo zie je eigenlijk... hoe je context op dit moment gevuld is. Dus 65.000 van de 200.000 tokens. En dan weet je ook meteen, oh... Ik gebruik blijkbaar Sonnet en ik heb blijkbaar 200.000 tokens. Je gaat even achter de schermen kijken. En dan denk je misschien, is het niet te nerdy? Hier wordt het echt leuk. Want als jij dan leert, stel dat je bijvoorbeeld alle documentatie van de NS-API hebt toegevoegd. En ineens loopt die uit context. Dan kan je kijken, heb ik net 150.000 tokens toegevoegd per ongeluk? Dat is eigenlijk een beetje dom. Ik had misschien alleen die drie pagina's van die API moeten toevoegen. Ja, dus jij kan zomaar zeggen... we gaan nu aan mijn recepten-app ook een reisplanner van de NS toevoegen. En dan gaat dat ding uit zichzelf zoeken naar... hoe werkt dat eigenlijk, een reisplanner toevoegen? En dan kom je er stiekem achter dat hij iets heeft uitgevoerd... wat dat hele blokjeschema laat vollopen. En dat is voor jou... iets om alarm over te slaan. Je betaalt ook per token. Kijk en nu. Wij hebben dan abonnementjes. Dus binnen het abonnement is het gewoon op op een gegeven moment. Maar je hebt ook gebruikers die ervoor kiezen. Ik wil door door. En dan moet je heel goed. Dus dit is wat hij. Bij elke keer als je iets vraagt. Dan neemt hij heel veel context mee. Ja, je zegt het goed hoor. Ja en nee, want Cloud Code ook wel weer slim genoeg is om hier en daar te zeggen. Hé, mijn contextwindow lijkt een beetje vol. Ik ga hem even samenvatten naar mezelf. Dus daarom is het een beetje een gekke tussentijd waarin we nu leven. Ik vraag me wel eens af. Ik heb een jaar geleden een soortgelijk verhaal verteld over Cloud Code. Maar laat ik het dan anders vragen. Welk probleem komen we dan tegen? Ja. Nou, op een gegeven moment ga je snel tegen... Kijk, je gaat bijvoorbeeld MCPs toevoegen. Omdat je denkt, ik vind het leuk dat ik kan praten met mijn administratie van Moneybird. Of mijn e-mail of mijn kalender. En dan ga je heel leuk, heel enthousiast allemaal MCPs aanzetten. Ja. Hoe doe ik

dat? In Cloud Code zit de mogelijkheid om extensies toe te voegen. Dat is best wel technisch nu nog. Dat gaan we buiten beschouwing laten. Je kunt, geen grap, dat heb ik gisteren ook getest. Dus ik ga toch antwoord geven op je vraag. Tegen Cloud Code zeg, kan je de MCP voor me toevoegen? Dat kan hij ook. Dan gaat hij hem zelf aan zichzelf toevoegen. Dus daar zijn we inmiddels ook aan. Dat hij een soort update doet van zichzelf. Maar mijn boodschap is vooral... more is not better. Je moet niet zomaar ineens al die leuke dingetjes. Het eerste wat mensen gaan doen is, oh, dan kan die daarbij en dan kan ik dat toevoegen en dan kan die dat lezen. Oh, hij kan naar mijn kalender en andere dingen. Maar je beseft je niet dat je je context window aan het vullen bent. En dat heeft te maken met hoe lang kan je cloud gebruiken? En als je voor tokens betaalt, wat gaat het je kosten? Het kan meestal uit de hand lopen. Dus een beetje manueel context engineering doen... kan in ieder geval een beetje begrijpen... wat gebeurt er nou eigenlijk achter de schermen. Dus doe soms slash context en kijk eens hoe het gaat. Of dat ding vol zit en dan wat? Dan kan je doen clear. Dan doe je even een frisse start. Sommige mensen starten gewoon het liefst nieuwe sessies. Maar je kan ook mid-sessie... Dan gaat hij opnieuw kijken... wat zijn we eigenlijk aan het doen. Dat voelt ook heel gek. Dan zegt hij, blijkbaar hebben we deze app. Ik ga toch even checken hoe alles zit. Dan kan je zeggen, is dat handig dan? Je kan natuurlijk ook een compact doen. Dat is eigenlijk het betere commando. Dat je zegt, luister... normaal doe je dat zelf... maar ik weet dat je dat eigenlijk beter nu kan doen. Het is toch een beetje duwen er tegen. Doe eventjes een samenvatting. Resumeer even. Hou de resume in je hoofd. En vergeet even alles daarvoor. En is dit iets wat je doet op het moment dat je merkt... Ik heb nu een langer gesprek. En ik merk dat hij eigenlijk niet precies... Hij lijkt steeds minder precies te reageren op wat ik vraag. Is dat dan typisch zo'n moment waarop je zo'n compact commando geeft? Ja, en ik doe het steeds minder. Want hij doet het zelf. Maar ik doe het nog wel eens. Omdat je toch... Als mensen het hele overzicht vaak nog wel hebt... van wat je aan het doen bent. Op een gegeven moment merk je gewoon... volgens mij moeten we even defragmenteren nu. Ja, ik snap het. Dan hebben we de kosten. Dus ook als je een abonnement hebt... goed om soms gewoon even te kijken... wat zouden mijn kosten zijn geweest? Want je ziet een soort virtuele kosten ook. Want je hebt een maand abonnement... dan ga je niet overheen. Maar hij kan je toch laten zien... dit zou je betaald hebben. En het heeft dan ook voor de mensen... die zich druk maken om het energiegebruik... tokens zijn energiegebruik. Dus in essentie zou je ook kunnen zeggen... wees een beetje... Leer toch een beetje efficiënt om te gaan met deze superkrachten, zou je kunnen zeggen. Over superkrachten, ik doe het even heel kort, want we hebben nog zes minuutjes tot vragen. Ik kan hier niet, ik wil dit meegeven aan beginners, want het is veel nu, de hoofden zitten vol. Maar ik ga het toch meegeven als een soort, dit is de logische volgende stap. Als het je lukt

om je eerste appjes te maken voor lokaal thuisgebruik. Is toch wel de... Ja, revolutie is een beetje een groot woord. Maar skills zijn toch wel een heel groot ding. En ze zijn best wel moeilijk uit te leggen. Want ze stellen technisch niet zoveel voor. Maar het is zo ontzettend krachtig. Gebruik jij zelf skills? Oké, dat is goed. Dan ga ik het je gewoon een beetje uitleggen. Skills zijn te gebruiken in Cloud Code. En in Cloud. De normale Cloud die je kent van cloud.ai. En inmiddels gaat OpenAI er ook mee aan de gang. In ieder geval, dat begreep ik. Skills zijn eigenlijk kleine stukjes prompt. Het zijn prompts die jij hebt verzameld. Noem het je eigen prompt library. Dus stel dat jij in jouw notities app van Apple... allemaal handige promptjes hebt verzameld. Ik ken mensen die dit hebben. Of op een plek, er een mapje met prompts. Dat zijn eigenlijk skills. Die skills, je kunt bijvoorbeeld... een van de standaard skills even concreet maken... die Cloud Code heeft, is de front-end developer. Jij kunt dus vragen in de frontend developers. Een markthelm bestandje. Dus een tekstbestandje met een klein beetje opmaak via tekens. Maar je kunt gewoon lezen. Gewoon tekst. Menselijke tekst. Daar staat eigenlijk. Jij bent een frontend developer. Een soort system prompt achtergevoel. Frontends moet je maken op deze en deze manier. Hele ideeën over hoe frontend werkt. Als je twijfelt. Zoek deze documentatie. Als je echt twijfelt. Gebruik deze MCP. Dus je kunt helemaal. Als je nog meer twijfelt. Hier heb je een stukje Python code. Dus je kunt. tekst invoeren en code en MCPs aanbieden. En wat kun je dan doen? Op het moment dat jij Cloud Code, deze skill aanwezig is en je vraagt Cloud Code om iets te bouwen, kan die zelf beslissen ik moet even die frontend skill erbij halen. Dan leest hij vlug even die prompt in, die voegt hij toe aan zijn hoofd, Windows het ware, en dan gaat hij verder. Of ik pauzeer nog even, want anders wordt het te veel. Pauzeer. Dan kan hij hem inlezen. Jij kunt zelf zeggen, gebruik de frontend skill maar. Omdat jij gewoon opinionated bent. Dus jij zou bijvoorbeeld zeker een.....redigeerskill maken voor teksten. En allemaal dingen die jij inmiddels hebt verzonnen.....ik kan je echt aanraden. Het zijn een soort rollen plus.....competenties, zou je kunnen zeggen. Skills. Het zijn gewoon een mapje. Echt geen grap. Mapje met punt MD. Het is technisch helemaal niks. Daarom namen heel veel mensen het ook niet echt serieus toen het uitkwam. Ja, dit is niks. Maar het is zo gestoordkrachtig. Want je creëert eigenlijk een prompt library... waar Claude zelf uit mag cherrypicken wanneer hij wil. En je kunt hem eigenlijk vragen om dat te doen. En je kunt skills combineren. En dan zijn we op de kerst op de taart... voor complexiteit van mijn verhaal vandaag. Dat zijn de subagents. Dat zijn eigenlijk extra cloud processjes die cloud zelf start. Dus je moet je voorstellen, cloud code is bezig om voor jou software te maken. Die wil een onderzoek doen, die start nog een cloudje erbij op de achtergrond. Maar dat doet hij dan met Haiku, een kleintje, goedkope, lekker snel. Omdat hij weet, dat is een prima taak om aan zo'n agent te geven. En die agent gaat bijvoorbeeld research voor jou doen. Dat is dus wat anders dan een skill, maar die kunnen

ook weer gebruik maken van skills. Waarom doe je dit? Het heeft ermee te maken dat je kosten wil besparen, snelheid wil, want je wil eigenlijk parallel aan elkaar dingen kunnen laten doen. En die sub-agents worden eigenlijk een soort, we hebben, als je zo'n marketingbureautje maakt, wat wij wel eens voor Eva hebben gedaan, voor de mensen die het niet hebben gezien, we hadden allemaal verschillende blokjes AI's aan elkaar gekoppeld, die met elkaar gingen praten. Dat doe je dan eigenlijk op de achtergrond binnen Classroom. En is dit dan het eerste voorbeeld waarbij die recepten, die food pairing app werd gemaakt? Is het dan dat er een subagent in de tussentijd het web op gaat om te kijken welke dranken passen ook echt bij welke gerechten? Dat is een subagent, maar in dat voorbeeld ging dat automatisch. Ja, dus je kan je afvragen. Nou, dus ik merk dus bij het voorbereiden van deze presentatie dat ik dacht... Hoeveel hiervan moet je eigenlijk doen? Moet je hier niet gewoon een beetje van afblijven of zo? Moeten we niet gewoon even wachten? Maar wat zou ik als ik zo'n... Want je zegt slash agents. Wat krijg ik dan als ik dat typ? Nou, je kunt zowel skills als agents... Klein dingetje even gezegd hebben. De slash commando's. Dus slash cost, slash compact. Die ik net allemaal heb laten zien. Alles werkt in de terminal. Nog niet alles werkt in de cloud app vanaf 6 januari. Even erbij. Als je ineens denkt, waar is Slash Agents? They're working on it. Mogelijk nu, vanochtend. Dus als je een Slash commando mist, zit die waarschijnlijk alleen in de terminal. Nog niet in die. Zo, de verf is nog nat. Als je met skills, je kunt aan cloud ook vragen. Wil je met mij een skill maken? Ja. Dan maakt hij hem, gaat hij met jou praten, defineert hij hem... schrijft hij hem zelf weg en voegt hij hem zelf aan zichzelf toe. Ik had het met zoveel mensen, hoe maak je jullie skills? Ik zei, hoezo maak skills? Ik schrijf ze helemaal niet zelf. Ik schrijf ze natuurlijk met klant. En ik merk dus dat ik... Ik zit te diep in de oude technologiegedachte... dat ik steeds nog denk dat ik zelf dingen moet gaan doen. Terwijl je eigenlijk gewoon veel slimmer moet zeggen... Ik heb hier een probleem. Zou je daar subagents voor gebruiken, Klad? En als we dat zouden doen, hoe zouden we die dan maken en noemen? Eigenlijk zeg je, je hoeft hier niet over na te denken. En als dat nu al niet zo is, dan over een paar weken. Ik geef hem dus aan het einde er even mee. Omdat, laten we zeggen, ben je al wat verder... of ben je over twee dagen verder en ben je mega enthousiast... Er is een rabbit hole. Hij is fantastisch. Je kunt nog veel meer doen. Voor de andere mensen zou ik zeggen. Ik denk dat Claude Coat dit in het eigen harnas. In de scaffold zelf. Op een gegeven moment gewoon lekker gaat oplossen voor je. Zullen wij naar de vraag gaan? Absoluut. Kjeld vraagt. Hoe verhoudt Claude zich tot Cursor? Dus ik zeg het even heel precies. Ik vraag niet Claude Coat. Maar hoe verhoudt Claude zich tot Cursor? Wat is Cursor? Cursor is ook een scaffold. Een van de jongste scaffolds. Namelijk een stukje software wat al bestond. Waarin we gingen programmeren. Het is een app die je downloadt van je website. Ja zeker. Een tekst editor voor programmeurs. Maar toen al. Dit voelt

allemaal een beetje gek. Maar in AI tijd lang geleden. Aangepast. Slim omgaan met het model. Dus een Superman pak. Cursor is ook zo'n Superman pak. Wat is het verschil? Dat Cursor... ook zelf modellen maakt. Een beetje verwarrend, maar Scourger werkt met alle providers. Dus je kunt en Cloud en GPT en Gemini... maar ook de Courser aangepaste fine-tunes erachter plakken. Wat is het verschil? De discussie is nu een beetje... Kijk, als je leuk aan het spelen bent... maakt het niet zoveel uit waar je heen en weer springt. Als jij binnen software aan het maken bent binnen een bedrijf... dan moet je wel een beetje ergens aan toewijden. Want je moet het hele team gaan opleiden erin. Ja. In mijn optiek is Cloud Code nu het meest krachtig. Want Entropic beheert de hele verticals, als ze dan zeggen. Dus en het model en de scaffold. En dat is gewoon iets wat Cursor ook probeert. Maar ze hebben gewoon niet het lab. Ze hebben niet die kracht eigenlijk om dat te doen. Zijn er mensen die net beginnen aan te raden om Cloud Cursor te gebruiken? Nee, dus niet Cursor. Niet Cursor te gebruiken, gewoon Cloud Code te gebruiken. Nee, dus ik zou dan zeggen, joh, als je op een gegeven moment uit het jasje groeit. Of denkt, ik wil eigenlijk andere modellen gebruiken die goedkoper zijn. Of meer voldoen aan mijn waarde. Of draai op mijn eigen computer. Ja. Dan kan je zeker keuren. Ik zie dat een beetje als een stap erna denk ik. Juist. Maar daar gaat deze masterclass niet over. Het is wel leuk dat mensen weten van het bestaan nu. Er is veel meer dan Cloud Code. Cloud Code was ook niet de eerste. Juist. Sander vraagt. Welke best practices zijn er om Cloud Code te sandboxen? Ik wil voorkomen dat de toegang heeft. Buiten een specifieke projectmap. Een onbedoelde systeemwijziging. Hoe bang moeten we zijn. Dat Cloud Code aan de haal gaat. Met je dagboeken. Moet je aan Wietse vragen? Wietse is natuurlijk wel bang. Ik draai alles in een virtual machine. Wat is dat? Een virtual machine is... Eigenlijk start je dan bijvoorbeeld Windows op je Mac. Of je Mac op je Mac. Oké. Het is een hele apart operating systeem eigenlijk. Wat je daarvoor draait. Ja, en dat bedoel... Dat kan je in hele technische kleine manieren doen. Dus ik moet zeggen, als je op Mac werkt... Je hebt een eigen virtualisatie framework, zoals dat heet. Je kunt natuurlijk Docker gebruiken, als je het wel mensen kent. Ik moet zeggen, er zijn best wel mooie tools... Maar even, moet dit? Want ik ben gewoon een gebruiker. Ik installeer Cloud, de app, op mijn computer. En selecteer die map keurig, zoals jij tegen mij hebt gezegd net. Moet ik echt dit allemaal draaien op een losse computer? Als het je bedrijfslaptop is, je moet het gewoon niet op je bedrijfslaptop doen. Punt 1. Is het je privé laptop, dan zou ik zeggen lees mee. Want Cloud vraagt de hele tijd netjes. Mag ik dit uitvoeren? Mag ik dat uitvoeren? Maar Cloud is wel bezig op je terminal. Die doet vaak een search. Als hij dan één slash vergeet, searcht hij wel door je hele document folder. Als jij niet meeleest. Op recente Macs krijg je dan constant pop-ups die zeggen... Cloud Code, wat wil je documenten? Wat vind jij ervan? Dus dan zeg ik nee, bijvoorbeeld. Ja, dat klinkt dus best veilig. Ja, op de Mac. Ik weet niet zo goed hoe de

sandboxing op Windows is. Maar goed, jouw vraag is... Hoe moet ik dit doen? Dus kies de machine waar gewoon weinig persoonlijke data op staat. Lees mee wat er gebeurt. Het gebeurt niet vaak dat er iets lekt, maar het is absoluut mogelijk. En ik moet wel even zeggen, er zijn ontzettend veel mooie sandboxing technieken en ook geautomatiseerd, maar dat is echt ver voorbij het beginnersniveau. Oké, Colin vraagt... bij softwareontwikkeling komen vaak veel meer dingen kijken... dan alleen maar bij een toeltje bouwen. In mijn ervaring bij grote bedrijven... kom ik regelmatig legacy applicaties tegen... die ooit als hobbyprojectje zijn gestart. Dus als een doeltje. Deze systemen zijn nu stiekem onmisbaar geworden voor het bedrijf... maar ook ongelooflijk moeilijk beheerbaar. Stel dat je toeltje nou echt verkoopbaar is... Hoe start je dan om het een soort van volwassen te maken? Versiebeheer, testen, veiligheid. Of is het handiger om hier vrijwel meteen mee te starten? Eigenlijk is dus de vraag, leuk zo'n toeltje. Maar als het handig werkt, dan wil ik ook dat het door meerdere mensen gebruikt kan worden, et cetera. Ik ga deze vraag beantwoorden vanuit de beginner tot en met, want het is een niveauotje hoger. Laat het bij beginner. Ja, ik kan de beginner ook helpen. Waar liep ik op een gegeven moment tegenaan? Het is misschien wel leuk om nog een aantal kanttekeningen. Ik had die app gewonshot, dus in één keer gemaakt, waarbij iemand een dagschema kan maken. De iPad app werkte eigenlijk helemaal, maar die web app had nog een foutje erin. En toen zei ik tegen hem, er zit een foutje in. Toen dacht ik, ja, dit schiet eigenlijk niet op. Hij moet gewoon zelf die app testen. Ik ben dan technisch genoeg om tegen Claude te zeggen... kan je niet zelf gewoon Chrome openen en rond gaan klikken? Test die app even. Dit noemen ze eigenlijk Full Circle Test Driven Development. Oftewel... Dit kan. Je kan aan Claude vragen, open een browser. Ja, één op de drie keer deed hij wat ik wilde. Anders zei hij, ik heb de browser geopend, succes. En dan stond die app daar en zei ik, ik heb Chrome voor je geopend. Toen zei ik, ja, maar nu moet jij gaan rondklikken. Ik zei, oké, ik ga de muis op je Mac overnemen. Nee, nee, nee, je moet Chrome besturen, want ik weet dat hij dat kan. Maar dit is nog niet standaard ingebouwd, zeg maar. Dus waarom zeg ik dit? Voor de beginners, wat je gaat merken is, als jij... Cloud ook de app kan laten testen. Rondklikken op de iPad. Rondklikken in de browser. Dit is niet standaard. Maar wel iets waar je naartoe moet werken. Dan krijg je eigenlijk wat ze noemen. Test Driven Development. Een oude manier van werken. Als in een klassieke goede manier van werken. Antwoord geven op die vraag. Ik heb Legacy Codebase. Het eerste wat er moet gebeuren is. dat er een testsuite gebouwd moet worden... om die hele oude codebase te testen. Want die testsuite houdt letterlijk in rondklikken... dingen aanmaken, dingen verwijderen, et cetera. Dat kan je met AI maken, die testsuite. Als AI ergens leuk voor is, leuk of goed... is het testsuites maken. Die testfeed, uiteindelijk druk je dan eens op een knop... en dan gaat hij gewoon zeggen... ik ga dit doen, ik ga dit doen... en dan zijn er van de 189 tests... 189 geslaagde applicatie werkt. Als

je dat helemaal hebt... dan kun je de applicatie stukje voor stukje... gaan vervangen met nieuwe stukjes code... en iedere keer weer de test laten runnen door AI. Dus je bent... Kijk, als jij een bestaande codebase gaat aanpassen... zonder die constant te blijven controleren of die werkt... dan gaat het je niet lukken. Dan moet je zelf heel de tijd rond gaan klikken. Dat is sowieso dom. Je wil eigenlijk dat die AI als opdracht heeft... Maak de code beter. Doe de tests. Als de tests falen, maak de code beter. En als je die cirkel maakt... Dit is ook wat heel veel... Een beetje van die vibe-codes online cool doen. Die zitten met hun smartphone. Twintig klats. Uren lang te draaien. Omdat zij die cirkel rond hebben gekregen. En dan... is het dus heel belangrijk als je zo'n refactor doet, dus opnieuw oude software naar nieuwe software brengen, de testsuite is als eerst. Je kan hem testen. En wat ik ook vaak bij softwarebedrijven zie, die zeggen refactor, we hebben nu zo'n goede testsuite, we laten AI de software erachter verzinnen, zodat de tests gaan werken. Dus dat klinkt misschien heel wauws, maar dat kan natuurlijk. Het klinkt wel een beetje als je hier het niveau van Cloud Code in je Cloud App ontstegen bent. Ja. Ja, goed. Meer over meer soort van nuances wat dat betreft. Wat is het verschil tussen Cloud Code en AI Studio van Google? En misschien dan ook maar Codex erbij trekken. Ik zou zeggen anti-gravity en Codex. Want ik moet zeggen, AI Studio heb ik nog niet gebruikt. Dus dat vind ik moeilijk om te zeggen. Maar dit zijn de grote pakketten. Google heeft anti-gravity. Cloud heeft cloud code. En Codex heeft open AI. Codex is ontzettend te vergelijken met cloud code. Die rennen vooraan de hele tijd. Je hebt gewoon Codex fans. Je hebt cloud fans. Je hebt veel mensen die gebruiken ze door elkaar. Maar goed, dat is wel geavanceerd. Anti-gravity is wat jonger. Ik heb het een aantal keer gebruikt. Ook leuk, want het is natuurlijk... En de scaffold is van Google of DeepMind. En het model. Dus wat is het verschil? Niet zo heel erg veel. Ik zou er vooral zelf mee werken om te kijken of het voor jouw doel werkt. Jeroen vraagt, bestaat er een community of forum voor beginnende Cloud Code gebruikers en Vibe Coders om ideeën uit te wisselen? Wie praat nog met andere mensen? Overhaupt vraag ik me dan af. Niet vanaf niche. Die heb ik niet zo te pakken. Klinkt als een type. YouTube is gewoon een goudmijn. Dat is echt goed dat je dit zegt. Alles wat ik weet is door YouTube tutorials te kijken. Die vaak hele klikbete titles hebben. En dat je echt denkt, dat kan niet veel zijn. En tien minuten later denk je, dit was wel heel veel. Er zijn zoveel Cloud Code Tutorials. En nu heb je de basis gehad bij jou. Ja, mijn doel was echt vandaag voor jullie om lekker te kunnen starten. En een beetje te laten zien. Zag je die iPad app net? Dat is toch bizar. Het is gewoon heel waardevol als je nog een uur hebt. Om bijvoorbeeld zo'n video van een uur. Waar iemand het gewoon van A tot Z iets ingewikkelders aan het bouwen is. Nu weet je wat er ongeveer gebeurt. En kijk dan naar iemand die iets ingewikkelds maakt met ervaring. Ja, en dan heb ik deze nog even in beeld gezet, ook als antwoord op die vraag. Er is een cloud code best practices van Entropic. Ik moet zeggen, dan

heb je nog geen mensen ontmoet, maar dan ben je wel jezelf aan het verdiepen. Je zou eens kunnen kijken, subreddits en dat soort zaken. Want er is wel veel hoor. Je hebt ook vibe coding subreddits. Misschien is dat wel gewoon het antwoord eigenlijk. Ja jongens, jullie hebben echt nog een miljoen vragen gesteld. Maar dit is het uur. Het spijt me. Ik ga jullie niet beter kunnen servicen wat we in een uur kunnen doen dan wat dit is. Wiets, ontzettend bedankt. Ja, heel erg leuk. Voor deze crash course. En jullie bedankt voor het kijken. In grote getalen. Nogmaals, we zijn hier weer op 5 februari. Zet dat in je agenda. En zo te zien hebben we de smaak te pakken. Met het geven van dit soort masterclasses. Dus we gaan ons best doen voor de volgende keer. Voor nu heel erg bedankt voor het kijken. Meedoen in de chat om elkaar te helpen. Blijf dat ook vooral doen. En je vragen in te sturen. En wij blijven ons uiterste best doen. Om zoveel mogelijk vragen te beantwoorden. Hopelijk tot de volgende AI Report. Die gaan we nu opnemen. En anders tot volgende maand. Doe, dank je.