

AI-agenten die het werk doen



# AI-agenten die het werk doen

*Hoe je ze maakt en aan het werk zet*

Floris Knol

Floris Knol

ISBN 978-94-0394-384-8

Omslagontwerp: Floris Knol

© 2026 Floris Knol

Eerste druk, 2026

Uitgegeven via Bookmundo

Niets uit deze uitgave mag worden verveelvoudigd of openbaar gemaakt zonder voorafgaande schriftelijke toestemming van de auteur.

Bij het maken van dit boek is gebruikgemaakt van AI-hulpmiddelen. De keuzes, de indeling en de formuleringen zijn van de auteur; de voorbeelden en de getallen komen uit werk dat hij zelf heeft gedaan.

# Inhoud

|                                           |    |
|-------------------------------------------|----|
| Voorwoord.....                            | 7  |
| Inleiding.....                            | 13 |
| Hoe de opdrachten in dit boek werken..... | 17 |
| De voorbeeldmap.....                      | 23 |

## DEEL I — ÉÉN DING GOEDKOOP MAKEN

|                                            |    |
|--------------------------------------------|----|
| 1 Eerst het gereedschap, dan het ding..... | 29 |
| 2 Wat wegviel, en wat niet.....            | 35 |
| 3 “Ja, maar jij bent programmeur”.....     | 45 |
| 4 Feiten, geen meningen.....               | 57 |
| 5 Bereikbaarheid boven waarde.....         | 67 |
| 6 Multipliers.....                         | 75 |

## DEEL II — EN DAN HONDERD TEGELIJK

|                                         |     |
|-----------------------------------------|-----|
| 7 En als je het duizend keer deed?..... | 87  |
| 8 Waarom doe je dat nog zelf?.....      | 95  |
| 9 Zichzelf controleren.....             | 103 |
| 10 Snelheid is geen comfort.....        | 119 |
| 11 Niet op schijf, niet gebeurd.....    | 127 |

### DEEL III — ALS HET MISGAAT

|    |                                       |     |
|----|---------------------------------------|-----|
| 12 | Fratsen .....                         | 137 |
| 13 | Fouten zijn je beste leraar .....     | 151 |
| 14 | Mijn gereedschap loog tegen me .....  | 163 |
| 15 | De tweede keer .....                  | 171 |
| 16 | Zoek de sleutel, niet de deuren ..... | 181 |
| 17 | Nul afhankelijkheden .....            | 191 |

### DEEL IV — HET WORDT EEN ORGANISATIE

|    |                                 |     |
|----|---------------------------------|-----|
| 18 | Het vliegwiel .....             | 201 |
| 19 | Wat een agent mist .....        | 211 |
| 20 | Grenzen, en wie tekent .....    | 219 |
| 21 | Verbeteren zonder release ..... | 229 |

### DEEL V — IN DE ECHTE WERELD

|    |                                      |     |
|----|--------------------------------------|-----|
| 22 | Toegang is schaarser .....           | 241 |
| 23 | Vastzitten is een ontwerpkeuze ..... | 251 |
| 24 | Meedoen in de formele wereld .....   | 263 |
| 25 | Als iedereen het kan .....           | 273 |
| 26 | Waar staan we nu .....               | 291 |

### TOT SLOT

|    |                          |     |
|----|--------------------------|-----|
| 27 | Waar gaat dit heen ..... | 305 |
|    | Termenlijst .....        | 319 |

# Voorwoord

Er is iets veranderd in het werk, en bijna niemand kan je precies vertellen wát.

Je hoort dat AI alles verandert. Je ziet voorbeelden die indrukwekkend zijn en waar je vervolgens niets mee kunt. Je leest berichten van mensen die het weten, cursussen die het beloven, en artikelen die na drie alinea's zeggen dat je vooral moet beginnen. En dan sta je maandagochtend voor je eigen werk en is er niets veranderd.

Dit boek is geschreven door iemand die het wél heeft gedaan, en het gaat over wat hij daarbij heeft geleerd.

Wat hier gebeurde

**Waar dit boek over gaat is een omgeving waarin agenten zelfstandig werk doen.** Niet één assistent die je helpt, maar meerdere die tegelijk draaien, met een opdracht, een grens en een plek waar ze opschrijven wat ze deden. En dat is op zijn beurt geen eindpunt: het is de stap naar een organisatie die op AI draait in plaats van er gebruik van maakt.

**Ik heb dat gebouwd**, vanaf niets, met wat er nu gewoon te krijgen is. Wat het verschil maakte was de manier van kiezen, en die vond ik onderweg.

Het ongewone eraan is niet de snelheid. Het is dat het elke dag makkelijker werd in plaats van moeilijker.

Dat is bij bijna alles andersom. Wat je bouwt wordt zwaarder: meer onderdelen, meer uitzonderingen, meer dingen die kapot kunnen, en dus minder tempo. Hier gebeurde het omgekeerde, en dat kwam niet door geluk of door talent. Het kwam door een klein aantal zetten die er op de dag zelf niets uitzagen.

### **Die zetten zijn wat dit boek je geeft.**

Een handboek van patronen die zich telkens herhaalden, met de voorbeelden erbij waaruit ze kwamen.

Alles is opgeschreven terwijl het gebeurde: wat er gedaan werd, waarom, wat het kostte, en wat er misging. Daar komen een stuk of tien patronen uit. **Die patronen zijn het boek; waar ze toevallig uit voortkwamen, is het voorbeeld** — en ze staan hier in de volgorde waarin je ze nodig hebt, niet in de volgorde waarin ze ontstonden.

**Dat onderscheid is belangrijk, want anders leest de helft van de lezers mee als toeschouwer.** Ik bouwde iets technisch, maar geen van de patronen gaat over techniek. Ze gaan over de volgorde waarin je dingen doet, over waar je tijd heen gaat, over wat je aan een machine kunt overlaten en wat niet, en over hoe je merkt dat je de verkeerde kant op werkt. Een boekhouder, een fotograaf, een advocaat en een aannemer komen ze alle vier tegen. Daarom staat bij elk patroon ook een voorbeeld dat niets met software te maken heeft.

**En je kunt het zelf nadoen.** Elk getal in dit boek komt uit werk dat echt is gedaan, en bij elk getal staat hoe het gemeten is. De opdrachten draai je op je eigen scherm, en dan zie je of je in de buurt komt. Dat is zeld-

zaam in dit onderwerp en het is met opzet: wat je zelf hebt zien werken, hoef je van mij niet aan te nemen.

Inclusief wat er misging. Er staat in dit boek een dag waarop ik drie keer achter elkaar de verkeerde conclusie trok, een moment waarop mijn eigen gereedschap tegen me loog, en een meting die bewees dat mijn manier van meten stuk was en niet het ding dat ik maat. Die staan er niet uit bescheidenheid. Ze staan er omdat daar het meeste te leren viel.

Waar dit heen loopt

**Wat begint bij één persoon met een leeg apparaat, eindigt bij iets dat zelf werk aanneemt in de echte wereld.**

Zo gezegd klinkt het als grootspraak, en dat snap ik. Maar het is letterlijk de opbouw van dit boek, en de stappen ertussen zijn geen van alle spectaculair. Het helpt om ze vooraf te zien, want dan begrijp je waarom het begin zo klein is.

**Het begint bij één ding goedkoop maken.** Eén taak sneller, één controle die zichzelf doet, één fout die je niet twee keer maakt. Dat is de eerste helft van het boek en het is gewoon handwerk.

**Daarna gaan er honderd tegelijk.** Niet omdat het indrukwekkend is, maar omdat de vraag verandert zodra er meer dan één is: wat je bij één met een blik ziet, moet bij honderd opgeschreven staan, en wat je bij één zelf doet, moet bij honderd zonder jou kunnen.

**Dan gaat er iets mis, en herstelt het zichzelf.** Iets dat merkt dat het vastloopt en stopt. Iets dat ziet dat dezelfde fout terugkomt en de instructie aanpast in plaats van de fout. Iets dat een grens heeft en er niet overheen gaat, ook niet als het uitkomt.

**En dan wordt het een organisatie.** Dat woord is niet groot bedoeld. Een organisatie is iets dat een doel heeft dat niemand elke ochtend opnieuw hoeft te geven, een budget dat op kan, rollen die van elkaar weten, en een geheugen. Zodra die vier er zijn, is het geen gereedschap meer dat op jou wacht.

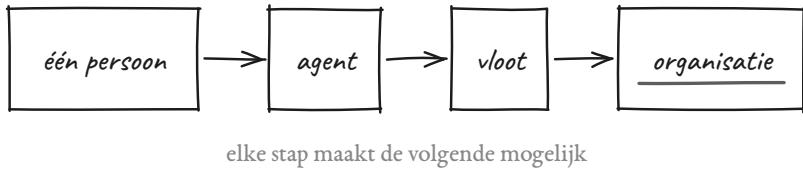
**En dan komt het deel dat bijna niemand noemt.** Zo'n organisatie doet dingen in de echte wereld. Ze koopt in. Ze meldt zich aan bij diensten en gaat daarmee verplichtingen aan. Ze huurt mensen in — want het idee dat er dan geen mensen meer aan te pas komen, is een fantasie: er is werk dat een mens moet doen en werk waarvoor een mens moet tekenen. Ze stelt doelen bij, en soms schaal ze zichzelf op.

En dus moet ze **meedoen in de formele wereld**. Er komt een accountant. Er komt een audit. Er is een norm waaraan een proces moet voldoen, en iemand die vraagt: laat maar zien wie dit heeft goedgekeurd, wanneer, en waarop hij zich baseerde.

**Dat is geen lastige bijkomstigheid maar de voorwaarde.** Een organisatie die niet kan laten zien wat ze gedaan heeft, is niet autonoom — die is onbeheerd. Het verschil tussen die twee woorden is precies het verschil tussen iets wat je in een bedrijf mag inzetten en iets wat je alleen thuis kunt uitproberen.

En het aardige is dat het dezelfde dingen zijn. Het vastleggen dat je nodig hebt om ervan te leren, is het vastleggen dat een auditor wil zien. De grens die voorkomt dat het geld op raakt, is de grens waar een controller naar vraagt. Wie het ene goed doet, heeft het andere er gratis bij — en wie het overslaat, moet het later alsnog bouwen op het moment dat het het slechtst uitkomt.

**Ik ben nog niet op het punt van een autonome organisatie.** Ik zit ergens in het midden van die opbouw, en het boek zegt precies waar. Maar de richting is duidelijk, en elke stap erheen is er een die je vandaag al kunt zetten.



De opbouw van dit boek: wat begint bij één persoon, eindigt bij iets dat zelf werk aanneemt.

## Hoe je het leest

**De vijf stappen hierboven zijn de vijf delen van het boek**, in precies die volgorde: één ding goedkoop maken, dan honderd tegelijk, dan wat er gebeurt als het misgaat, dan wordt het een organisatie, en dan de echte wereld. Daarna volgt nog één hoofdstuk dat verder kijkt dan waar ik zelf sta — het enige dat nog geen praktijk is.

Elk hoofdstuk daalt minstens één keer af naar iets dat je vanavond kunt uitproberen. Het staat gewoon in de tekst, met erbij waarom het werkt en waar het misgaat. Je kunt eroverheen lezen; je kunt ze ook allemaal doen, en dan heb je aan het eind iets gebouwd terwijl je las.

En elk hoofdstuk sluit met één regel: **wat dit goedkoper maakte**. Die regels stapelen. Lees ze een keer achter elkaar terug en je ziet waar het boek eigenlijk over gaat.



De vijf delen staan op elkaar: elk deel maakt het volgende mogelijk.

Waarom nu

Omdat je nog niet te laat bent.

Dat is geen verkooppraatje maar een waarneming. De dingen die dit mogelijk maken zijn er nu allemaal, ze zijn goedkoop, en bijna niemand gebruikt ze samen. Dat is precies het moment waarop iemand die het wél doet een voorsprong opbouwt die niet in te halen is met geld — want de voorsprong zit in wat je op elkaar hebt gestapeld, en stapelen kost tijd die niemand kan overslaan.

Over twee jaar is dit gewoon. Nu nog niet.

# Inleiding

Ik ben programmeur. Dat is de makkelijkste manier om me weg te zetten, dus laat ik er meteen iets naast leggen.

Ik heb ook jarenlang mensen begeleid naar werk. Mensen zonder betaald werk, mensen met autisme, mensen die lang buiten het arbeidsproces stonden, mensen voor wie de gewone route niet werkte. Dat is ander werk dan software bouwen — andere woorden, andere uitkomsten, en niemand verwacht de twee.

In dat werk leerde ik vier dingen af en aan, en geen ervan ging over computers. **Vooroordelen helpen niet**, want wie met een dossier in zijn hoofd binnenkomt ziet bevestiging van wat hij al dacht. **Opbouwen wél**, stap voor stap, omdat een stap die mislukt alleen iets vertelt als het er één was. **Herhalen en stapelen**, want wat één keer lukt kan toeval zijn en wat drie keer lukt is een bodem om op te staan. En **systematisch en voorspelbaar maken** — dezelfde volgorde, dezelfde woorden — omdat je daarmee alle ruis weghaalt die niets met de zaak te maken heeft.

Er was er nog één, en die bleek later de belangrijkste: **geef feiten, geen aannames en geen meningen**. *"Hij is niet gemotiveerd"* is een eindpunt. *"Hij kwam de afgelopen drie keer op dinsdag niet"* is een beginpunt. Het eerste sluit af, het tweede opent.

Toen ik met AI ging werken, hoefde ik niets nieuws te leren. Ik deed wat ik al deed, en het werkte meteen.

**Dat verbaasde me meer dan het had moeten doen.** Een model heeft geen voorgeschiedenis met je, kan niet raden wat je bedoelt, en leest niet tussen de regels door. Alles wat je niet als feit levert, vult het zelf in. Precies dezelfde beperking als in een kamer met iemand die je voor het eerst ziet — en dus precies dezelfde werkwijze.

Zo is dit boek ontstaan. Niet vanuit een idee over AI, maar vanuit de vaststelling dat drie werelden die niets met elkaar te maken hebben, dezelfde vragen stellen. De patronen hierin heb ik niet bedacht; ik heb ze herkend. Dat maakt ze niet nieuw. Het maakt ze bruikbaar op plekken waar je ze niet zou zoeken.

Wat er wél nieuw aan is, is niet het idee. **Het is dat er nu iets bestaat dat ze uitvoert terwijl jij er niet bij bent.**

Dit boek gaat over wat dat verandert, en wat je moet doen om het te laten kloppen.

**En daarom staat er geen enkele truc in.** Wat een paar maanden werkt en daarna onjuist is, leert je niets — je weet nog steeds niet wáárom het werkte toen het werkte.

Wat er wel in staat zijn concepten en patronen. Die gaan over de volgorde waarin je dingen doet, over waar je aandacht heen gaat, over wat je kunt overlaten en waaraan je dat ziet. Zulke dingen zijn niet gebonden aan een programma, een leverancier of een versie. **Ze golden voordat dit gereedschap bestond, ze gelden nu, en ze gelden nog steeds als alles wat je vandaag gebruikt vervangen is door iets beters.**

Dat is de belofte van dit boek, en het is de enige die ik durf te doen: je hebt er over een paar jaar nog steeds iets aan, in welke omgeving je ook werkt.

**En toch daalt elk hoofdstuk één keer af naar iets dat je vanavond kunt doen.** Dat lijkt in tegenspraak met de alinea hierboven en dat is het niet — maar het verdient een eigen uitleg, en die staat hierna. Wie liever meteen begint slaat hem over en gaat naar hoofdstuk 1.



## Hoe de opdrachten in dit boek werken

Dit hoofdstuk staat vooraan omdat het over de vorm gaat en niet over de inhoud. Je kunt het overslaan en meteen aan hoofdstuk 1 beginnen; je komt dan alleen een paar keer iets tegen waarvan je je afvraagt waarom het er zo uitziet.

In het kort: elk hoofdstuk dat zich ervoor leent daalt één keer af naar iets dat je vanavond kunt doen. Wat hieronder staat is waarom dat er is, hoe je het gebruikt, waar de voorbeelden vandaan komen, en wat er aan veranderd is voordat het in dit boek kwam.

Wat een uitgewerkt voorbeeld doet

**Een uitgewerkt voorbeeld is de kortste weg van een concept naar iets dat op jouw scherm gebeurt**, en juist daardoor blijft het concept hangen. Je hebt het één keer zien werken; de tweede keer verzin je hem zelf.

Dus daalt elk hoofdstuk dat zich ervoor leent één keer af naar iets dat je vanavond kunt doen: wat je doet, wat je letterlijk typt, wat je terugkrijgt, en waar het misgaat. Reken op vijf tot twintig minuten. Je kunt ze overslaan en het boek klopt nog; je kunt ze allemaal doen en dan heb je aan het eind iets gebouwd terwijl je las.

**De voorbeelden zijn geschreven voor één programma, en dat is Claude Code.** Niet omdat het het beste is en niet omdat het moet, maar omdat er iets moet staan dat je kunt overtypen. Een voorbeeld dat zegt

*"vraag je AI om zoiets"* is geen voorbeeld maar een uitvlucht: het schuift het enige moeilijke stuk — precies opschrijven wat je wilt — terug naar de lezer.

**Bij elke andere aanbieder werkt het hetzelfde, met dezelfde woorden.** Wat deze voorbeelden van je vragen is nooit een knop, een menu of een instelling; het is een opdracht in gewone taal, en gewone taal is nu juist wat al die programma's gemeen hebben. Wáár je het intypt verschilt en verandert bovendien elk halfjaar. Wát je intypt niet.

**En dit is geen inleiding in het gebruik van zo'n programma.** Dat je een opdracht kunt typen en antwoord krijgt, wordt hier verondersteld; daar zijn handleidingen voor en die zijn beter dan wat ik erover zou schrijven. Wat hier begint is de stap dáárna, en dat is een andere stap dan de meeste mensen denken.

**Die stap is: van iets dat antwoordt naar iets dat wérkt terwijl jij er niet bij bent.** Niet een gesprek waarin jij elke beurt de volgende zet doet, maar een opdracht die wordt uitgevoerd, gecontroleerd en afge maakt zonder dat jij ertussen zit. Het verschil tussen die twee is niet gra dueel. Zodra jij niet meer elke beurt meekijkt, moet alles wat jij stilzwij gend deed ergens anders vandaan komen: wat er precies moet gebeuren, waar het ophoudt, waaraan je ziet dat het klopt, en wat er gebeurt als het niet klopt.

**Dat is waar dit boek over gaat, en waar de voorbeelden je heen bren gen.** Het eerste is klein en doe je nog met je hand op de knop. Elk vol gende haalt een stukje van jou uit de lus — tot er iets staat dat een op dracht aanneemt, uitvoert, zichzelf nakijkt en pas terugkomt als het klaar is of als het ergens niet uit komt.

**Alle voorbeelden gaan over hetzelfde bouwbedrijf.** Eén wereld die meegroeit: opdrachten, opdrachtgevers, personeel met certificaten, materieel met keuringen, facturen, een agenda, aanbestedingen — en twaalf regels waar het zich aan hoort te houden. Klein genoeg om erin te kijken, echt genoeg om ergens op mis te gaan. Het eerste voorbeeld laat je die map beschrijven; daarna werkt elk voorbeeld erin. Aan het eind heb je één ding opgebouwd in plaats van twintig losse proefjes.

**Die map maak je zelf, en dat kost één opdracht.** Hoe dat gaat en waar je op let, staat in het volgende hoofdstuk — daar begint het werk.

**En alles wat je in die voorbeelden ziet terugkomen, is er echt uitgekomen.** Elk voorbeeld in dit boek is uitgevoerd in die map, met precies de opdracht die erbij staat. Wat er onder *wat er terugkwam* staat, is wat er terugkwam — niet een nette versie ervan, niet iets dat ik heb overgetypt omdat het beter stond.

**Twee dingen heb ik er wél aan gedaan, en het is eerlijker om ze te noemen.** Ik heb de stukjes uitvoer **ingekort** zodat ze op een boekbladzijde passen: nooit meer dan acht regels en nooit breder dan de kolom. En ik heb er de paden en de datums van mijn eigen machine uit gehaald, omdat die bij jou toch anders zijn. **Geformuleerd is er niets.** Waar een antwoord onhandig was, staat het onhandig; waar het fout was, staat de fout er nog — en in een paar gevallen is die fout het punt van het hoofdstuk geworden.

**Eén ding heb ik niet echt gedaan, en dat hoor je te weten.** In de laatste hoofdstukken reiken de voorbeelden naar buiten: een e-mail aan een opdrachtgever, een melding aan de gemeente, een boeking in een boekhouding. Die opdrachten zijn echt uitgevoerd, maar in een oefenopstelling — er is geen mail verstuurd en er is met geen enkel systeem van ie-

mand anders gepraat. Wat er terugkomt is wat zo'n ontvanger zou antwoorden, inclusief de keren dat hij níet antwoordt.

Bij jou gaat het wél echt de deur uit, en dat is precies waarom die hoofdstukken over grenzen en handtekeningen gaan.

Dat is met opzet. Een voorbeeld dat is bijgewerkt tot het mooi stond, zou precies de ervaring wegpoetsen die je zelf gaat hebben.

Kopieer ze niet. Doe ze.

Hier wil ik één ding met nadruk zeggen, omdat het het verschil maakt tussen een boek dat je hebt gelezen en een boek waar je iets aan hebt gehad.

**De opdrachten in dit boek zijn geen recepten.** Ze staan er zodat je ziet hoe zo'n opdracht eruitziet en wat hij oplevert — niet zodat je ze overtuikt en dan verder leest. Wie ze alleen overtuikt, krijgt iets soortgelijks als wat er in het boek staat en leert daar bijna niets van, want het klopte al.

En let op: **je krijgt sowieso niet letterlijk wat er in dit boek staat.** Vraag twee keer hetzelfde en je krijgt twee keer iets soortgelijks, in andere woorden en soms in een andere volgorde. Dat is geen storing maar hoe dit werkt — en het is de tweede reden om niet te vergelijken maar te proberen. Wat moet kloppen is de uitkomst, niet de bewoording.

**Verander ze.** Zet er je eigen regel in, je eigen map, je eigen getal. Vraag om iets anders dan wat er staat. Laat iets weg en kijk wat er dan gebeurt. **Dat is waar het leren zit**, en het is ook precies wat er gebeurt zodra je het op je eigen werk toepast: daar staat nooit hetzelfde als hier.

**Daarom eindigt bijna elk voorbeeld met een tweede ronde.** Eerst de opdracht, dan wat er terugkwam, dan wat er niet goed aan was, en dan

een bijgestelde opdracht met een nieuw antwoord. Die tweede ronde is er niet voor de volledigheid. Hij laat zien dat de eerste poging zelden goed is, en dat bijsturen geen teken is dat je het niet snapt maar de manier waarop dit werkt.

Doe er één. Verpruts hem. Stel hem bij. Dan heb je het te pakken.

En ze veranderen van soort onderweg

**Wat je doet verandert door het boek heen, en dat is de hele beweging die dit boek beschrijft.**

**In het begin typ je een opdracht en lees je het antwoord.** Jij zet elke keer de volgende stap. Dat is waar bijna iedereen blijft hangen, en het is ook prima — er valt veel te winnen met alleen dat.

**Even verderop leg je vast wat je net deed,** zodat je het niet elke keer opnieuw hoeft uit te leggen. Je bent dan niet meer aan het uitvoeren maar aan het inrichten, en je merkt dat je sommige dingen nooit meer typt.

**Daarna zet je dingen klaar die zichzelf aanroepen.** Een controle die draait als er iets is bijgekomen. Een grens die eerst tegenhoudt en dan pas vraagt. Jij bent er op dat moment niet bij, en dat is het punt: de volgende stap wordt niet meer door jou gezet maar door de situatie.

**En aan het eind zet je geen opdrachten meer klaar.** Dan loopt er iets dat je ooit hebt opgezet. Je checkt in, je kijkt wat er is gebeurd terwijl je er niet was, en af en toe stuur je iets bij. Meer niet. **Het loopt door.**

Daarom ziet een voorbeeld in het laatste deel er anders uit dan een voorbeeld in het eerste. Daar staat geen *typ dit* meer, maar *kijk wat er is gebeurd*. Dat is niet omdat er niets meer te doen is — het is omdat het doen ergens anders is komen te liggen.

Dat zeg ik hier één keer, en daarna niet meer bij elk voorbeeld. Een voorbehoud dat honderd keer terugkomt, wordt honderd keer overgeslagen.

# De voorbeeldmap

Alle opdrachten in dit boek spelen in dezelfde map. Voordat je aan hoofdstuk 1 begint, moet die map er zijn — en dat kost één opdracht.

## Waarom een bouwbedrijf

Het gaat om werk met afspraken erin, en dat heeft elke organisatie — een school, een cateringbedrijf, een administratiekantoor. Een bouwbedrijf is in dit boek een concreet voorbeeld.

**Daarin zit een web dat administratie stukmaakt.** Opdrachten verwijzen naar opdrachtgevers, opdrachtgevers naar facturen, facturen naar uren, uren naar mensen, mensen naar certificaten met een geldigheidsdatum. Dat is precies wat een agent kan nalopen en jij niet.

**En je hebt er geen verstand van nodig.** Iedereen snapt dat een certificaat geldig moet zijn voordat je de steiger op gaat. Vervang bouwbedrijf door wat jij doet — de rest van het boek leest hetzelfde.

## De opdracht

Maak een lege map, open je agent erin, en typ dit:

*Maak in deze map de administratie van een bouwbedrijf met vijf man vast personeel. Gebruik mappen voor opdrachten, klanten, personeel met certificaten, materieel met keuringen, facturen, urenstaten, offertes, inkoop, onderaannemers, agenda en aanbestedingen — een stuk of dertig in totaal, met een paar documenten elk. Schrijf alles als markdown met velden bovenaan. Zet er een `regels.md` bij met twaalf regels waar de administratie zich aan hoort te houden: nummering van opdrachten, geldige certificaten bij het werk dat iemand doet, keuringen van materieel, een grens voor meerwerk, en zo verder. Laat elke regel precies één keer overtreden worden, verspreid over de map, zonder dat erbij staat waar.*

Reken op een paar minuten. Je krijgt geen voortgangsbalk; je ziet mappen verschijnen en daarna een samenvatting van wat er is gemaakt.

Waar je op let als hij klaar is

Vier dingen, en ze zijn allemaal te tellen:

- **ongeveer dertig mappen** op het eerste niveau, met de elf uit de opdracht erbij
- **rond de honderd documenten**, verdeeld over die mappen
- **`regels.md` met twaalf regels**, genummerd
- **velden bovenaan elk document** — een opdracht heeft een nummer, een klant, een bedrag en een datum, geen lopende tekst waar je die uit moet vissen

Klopt dat, dan kun je verder. De twaalf overtredingen ga je niet zoeken: dat is het werk van hoofdstuk 1 en verder, en de agent heeft ze met opzet niet opgeschreven.

Wat je terugkrijgt is niet wat ik heb

Jouw map ziet er anders uit dan de mijne. Andere namen, andere bedragen, andere volgorde, en waarschijnlijk een paar mappen die ik niet heb — of andersom.

**Dat is goed, en het is de reden dat dit werkt.** De patronen in dit boek hangen niet aan een bestandsnaam. Ze hangen aan de vorm: er staan afspraken vast, er is iets dat zich er niet aan houdt, en er is een manier om dat te vinden zonder alles met de hand na te lopen.

Wat wel afwijkt zijn de **getallen in de uitkomsten**. Waar in dit boek staat dat een controle er negen vond, vindt die van jou er misschien zeven of elf. Vergelijk dus niet het getal maar de vorm: kreeg je een lijst met vindplaatsen, of kreeg je "ziet er goed uit"? Dat verschil is waar het in hoofdstuk 4 over gaat.

Als het misgaat

Er zijn drie manieren waarop dit niet meteen lukt, en geen ervan vraagt om opnieuw beginnen.

**Te weinig mappen.** Sommige agenten maken er tien en stoppen. Vraag om aanvulling: *"maak er nog twintig bij, dezelfde opzet"*. Aanvullen is goedkoper dan overdoen, en het is meteen de eerste keer dat je merkt hoe zo'n gesprek werkt.

**Geen overtredingen.** Een agent die nette administratie maakt, heeft het laatste deel van de opdracht genegeerd — dat gebeurt, want het is een rare vraag. Zeg het dan expliciet: *"voeg nu per regel uit regels.md precies één overtreding toe, verspreid, en schrijf nergens op waar"*.

**Lopende tekst in plaats van velden.** Krijg je documenten die lezen als een brief, vraag dan om de vorm: *"zet bovenaan elk document de velden op een eigen regel, zoals bedrag: 1850"*. Zonder die vorm kan een controle er later niet op rekenen.

En als je het toch liever overdoet: gooi de map leeg en typ de opdracht opnieuw. Er gaat niets verloren — dit is precies het soort werk dat goedkoop hoort te zijn, en dat is waar hoofdstuk 10 over gaat.

**Dit is de enige stap in het boek die je maar één keer doet.** Alles daarna bouwt erop voort: elk voorbeeld werkt in deze map, en aan het eind staat er iets dat zichzelf nakijkt.

DEEL I

# Één ding goedkoop maken



## i. Eerst het gereedschap, dan het ding

Het is twee minuten over acht 's avonds. Er is nog niets.

Vier minuten later staan er vier dingen op schijf. Geen van de vier is het ding dat gebouwd moest worden.

Er staat **gereedschap** om het mee te maken. Er staat een **manier om te controleren** of wat er gemaakt wordt ook klopt. Er staat een **lijst van wat er nog moet**, met de volgorde erin. En er staan **voorbeelden**: een paar kleine dingen die al werken, om van af te kijken.

Het ding zelf staat er niet. Dat komt morgen.

Wat daar gebeurde

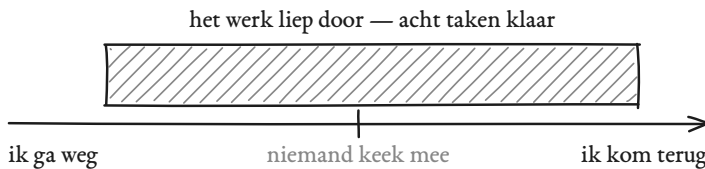
Iemand die iets wil maken, maakt het. Dat is niet gek: je hebt een idee, je gaat aan de slag, en het gereedschap komt wel als je het nodig hebt. Zo doet bijna iedereen het, en zo had ik het zelf ook altijd gedaan.

Wat er die avond gebeurde was iets anders. De eerste zet was niet *beginnen* maar **zorgen dat beginnen goedkoop wordt**.

Dat kostte een avond en leverde die avond niets op. Er was aan het eind van de dag niets te laten zien. Er was alleen een plek waar iets maken minder tijd zou gaan kosten dan ergens anders.

Dat is het eerste patroon van dit boek, en het is het patroon waar alle andere op rusten:

**Voordat je het ding maakt, maak je de dingen die het ding goedkoop maken om te maken — en goedkoop om te controleren.**



Wat er gebeurde tussen het moment van weglopen en het moment van terugkomen.

Waarom die avond er überhaupt was

Er hoort een reden bij, want anders is dit een avond met een goede gewoonte en niet het begin van iets.

**De aanleiding was de omvang.** Niet van het project — van het gereedschap. Een moderne ontwikkelstapel is zo groot geworden dat er geen korte lus meer in past. Er zit zoveel tussen het moment dat je iets verandert en het moment dat je weet of het klopt: bouwen, laden, wachten, starten, afhankelijkheden die iets anders willen. Voor een mens is dat vervelend. Voor een machine die het honderd keer achter elkaar doet, is het fataal.

**En dat is het hele probleem in één zin: bij zo'n stapel kán er geen korte terugkoppeling voor een machine ontstaan.** Alles wat er in dit boek staat over proberen, meten, opnieuw proberen en zichzelf laten nakij-

ken, rust op een lus die kort genoeg is om honderd keer te draaien. Is hij dat niet, dan is het allemaal theorie.

Die avond ging dus over het inkorten van die lus, en wat eruit kwam was wat er nodig was om hem kort te krijgen.

**Verwachtte ik dat het zou lukken?** Ja, en dat is geen bravoure. Ik had het de maanden ervoor al zien gebeuren in het klein — genoeg om te weten dat het zou moeten kunnen. Niet *dat het zou lukken* maar *dat het moest kunnen*, en dat is precies het verschil tussen een gok en een verwachting die op iets rust. Dit is dezelfde beweging als het turven op dat kladblaadje uit hoofdstuk 3: eerst meten, dan zeker weten.

Dit gaat niet over software

Het klinkt misschien alsof dit iets is voor mensen die programma's schrijven. Het is precies andersom: het valt daar alleen sneller op, omdat je er alles kunt meten.

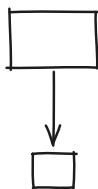
Een fotograaf die tweeduizend opnamen per opdracht maakt, kan die allemaal één voor één bekijken. Of hij kan eerst een halve dag besteden aan het vastleggen van zijn belichting en zijn bewerking, en daarna begint elke opdracht op tachtig procent. Die halve dag levert die dag niets op.

Een advocaat kan elk contract opnieuw schrijven. Of hij kan een verzameling clausules aanleggen die hij hergebruikt, en dan is het schrijven van een contract iets anders geworden dan het daarvoor was.

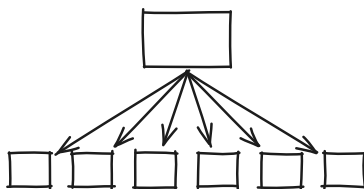
Een docent kan elk jaar dezelfde uitleg geven en elk jaar opnieuw merken waar leerlingen struikelen. Of hij schrijft één keer op wáár ze struikelen, en verandert de uitleg daar.

**In alle drie de gevallen is de zet die telt de zet die op die dag niets oplevert.** Dat is geen toeval; dat is de definitie. Een zet die meteen iets oplevert, levert meestal alleen dát op.

één taak tegelijk



veel taken tegelijk



Eén persoon, één taak tegelijk — en dezelfde persoon met veel taken tegelijk.

## Zelf doen · Vraag wat er eigenlijk in staat

**Wat je doet.** Open de voorbeeldmap en vraag wat erin zit. Niets bouwen, niets veranderen — alleen kijken. Dit is de kortste versie van wat dit hoofdstuk beschrijft: je typt één zin en je krijgt iets terug waar je anders een ochtend voor had gezeten. Reken op twee minuten.

TYP DIT

Dit is de administratie van een bouwbedrijf. Beschrijf in tien regels welke bedrijfsprocessen je hier terugziet, en noem daarna drie dingen die je opvallen.

**Wat je ziet.** Een opsomming van processen die je zelf nooit zo op een rij had gezet — van acquisitie en calculatie tot kwaliteitsborging en nazorg — en drie opmerkingen over wat eruit springt. **Let op de derde opmerking.** De eerste twee zijn meestal wat je al wist; de derde is regelmatig iets waar je nooit naar had gekeken.

## WAT ER TERUGKWAM

```
koppelingen 17  opdrachten 15  uitvoering 14
urenstaten 12   agenda 11    ontwikkeling 11
kam 11          facturen 9    calculaties 8
magazijn 8      personeel 8
werkvoorbereiding 8
... 49 mappen, 254 bestanden
opvallend: 1 gat in de opdrachtnummering
opvallend: 1 opdracht zonder materiaal
```

**Waar het misgaat. Vraag om een aantal regels.** Zonder "in tien regels" krijg je een verslag van twee kantjes, en dan lees je alsnog alles. **Vraag er iets bij dat een oordeel vraagt.** "Beschrijf wat erin staat" levert een inhoudsopgave op; "noem drie dingen die je opvallen" dwingt tot kiezen, en in dat kiezen zit de waarde. **En geloof de opsomming niet zomaar.** Wat hier opgesomd wordt is wat er in de bestandsnamen staat, niet wat er werkelijk gebeurt — dat verschil is precies waar hoofdstuk 14 over gaat.

## EN DAN ZO

Noem bij elk van die drie opvallende punten het bestand waar je het uit haalt, zodat ik het zelf kan nakijken.

## EN DAN DIT

```
gat in de nummering
-> opdrachten/, 15 bestanden, NLOP-2026-005 ontbreekt
opdracht zonder materiaal
-> opdrachten/NLOP-2026-003.md
opdrachtgever onbekend
-> opdrachten/NLOP-2026-006.md
```

## Het ongewone deel

Wat er later stond had ik in mijn eentje nooit kunnen bouwen als ik gewoon was begonnen. Maar dat is niet het interessante getal.