

JarkoMes.

JARKOMES BV · PRAKTIJKGIDS

Van AI-pilot naar productie

De vragen die je beantwoordt vóór je bouwt, en waarom de meeste AI-projecten precies daar stranden.

De meeste AI-projecten halen productie niet

Niet omdat het model tekortschiet. De modellen zijn al een tijd goed genoeg voor het soort werk waar bedrijven ze op zetten: gegevens overzetten, dossiers controleren, rapporten samenstellen, signalen opvangen.

Ze stranden op organisatie. Op de vraag wie eigenaar is van het proces eronder. Op wat er gebeurt als de agent het fout heeft. Op het feit dat degene die het bouwde het bedrijfsproces niet kende, of dat degene die het bedacht de techniek niet kende.

Deze gids bevat wat ik gebruik voor ik aan een bouw begin. Geen theorie, geen leverancierspraat. De vragen die het verschil maken tussen een agent die na drie maanden stilvalt en een die er over twee jaar nog staat.

Je kan dit document gebruiken zonder mij. Dat is de bedoeling. Als je het daarna toch samen wil doen, staat mijn nummer achteraan.

Drie vragen die niemand vooraf beantwoordt

Ik heb genoeg pilots van dichtbij gezien om te weten waar ze omvallen. Vrijwel altijd op één van deze drie.

01 **Wie is eigenaar van het proces?**

Niet van het AI-project — van het proces zelf. Als de agent morgen wegvalt, wie doet het werk dan? Als dat antwoord "dat zoeken we dan uit" is, is er geen eigenaar, en dan is er ook niemand die merkt dat de agent iets fout doet.

02 **Wat gebeurt er als hij twijfelt?**

Een agent die alleen het goede pad kent, is een demo. Er moet een drempel zijn waarboven een mens beslist, en die drempel moet vóór de bouw vastliggen, niet erna. Anders wordt hij in productie ad hoc bepaald, meestal door wie toevallig de fout ontdekt.

03 **Wie kan hem aanpassen als het bedrijf verandert?**

Processen veranderen. Als niemand in je eigen team begrijpt hoe de agent werkt, heb je geen automatisering gekocht maar een afhankelijkheid. Overdracht is geen nazorg, het is onderdeel van de opdracht.

Twaalf vragen vóór je bouwt

Beantwoord deze twaalf en je weet of je klaar bent om te bouwen. Kan je er drie of meer niet beantwoorden, dan is het proces zelf nog niet scherp genoeg en gaat automatiseren het probleem vergroten in plaats van oplossen.

Het proces

- ☐ Kan je het proces in vijf stappen op papier zetten, zonder "en dan kijkt iemand ernaar"?
- ☐ Hoeveel keer per week gebeurt het, en hoeveel tijd kost het per keer?
- ☐ Wat kost een fout vandaag, in geld of in tijd om recht te zetten?

De beslissingen

- ☐ Welke beslissingen mag de agent zelf nemen?
- ☐ Welke moet hij voorleggen, en aan wie precies?
- ☐ Welke mag hij nooit nemen, ook niet als hij zeker lijkt?

De data

- ☐ Uit welke systemen moet hij lezen, en heb je daar leestoegang voor?
- ☐ Waar schrijft hij naar, en is dat omkeerbaar?
- ☐ Zit er persoonsgegevens of gevoelige bedrijfsdata in, en mag die het pand uit?

Het vangnet

- ☐ Waar zie je wat de agent gedaan heeft, zonder het hem te vragen?
- ☐ Hoe merk je dat hij stilstaat, en binnen hoeveel tijd?
- ☐ Wie krijgt het signaal als er iets misgaat, en op welk kanaal?

Wat “in productie” eigenlijk betekent

Dit is het woord waar de meeste misverstanden in zitten. Een demo die op je laptop werkt is niet in productie. Een script dat iemand handmatig start is niet in productie. Een proof of concept die op een testomgeving draait al helemaal niet.

Hij draait zonder dat iemand hem start

Op een schema of op een gebeurtenis, op infrastructuur die niet iemands laptop is.

Hij overleeft een fout

Een systeem dat even onbereikbaar is, een veld dat leeg blijkt, een formaat dat verandert. Hij stopt, meldt, en pakt de draad weer op.

Je ziet wat hij deed

Logging die een mens kan lezen, niet alleen een stacktrace. Achteraf moet je een beslissing kunnen reconstrueren.

Iemand is verantwoordelijk

Een naam, geen afdeling. Die persoon krijgt de escalaties en mag de agent uitzetten als het moet.

Je team kan hem aanpassen

Documentatie, toegang tot de code, en iemand die de logica begrijpt. Zonder dat is elke wijziging een nieuw project.

Mijn eigen regel: wat ik oplever draait in productie of het is niet af. Een agent die alleen in een demo werkt is geen agent, dat is een presentatie.

Processen die zich hier goed voor lenen

Niet alles moet een agent worden. Dit zijn de patronen waar het in de praktijk het snelst rendeert: repetitief, regelgebaseerd genoeg om af te bakenen, en vervelend genoeg dat niemand het graag doet.

Reconciliatie en controle

Facturatie tegen CRM, betalingen tegen boekhouding, contract tegen levering. Saai, repetitief, foutgevoelig.

Monitoring en opvolging

Signalen over systemen heen opvangen, verbanden leggen, en escaleren voordat iemand het zelf merkt.

Onboarding en klantopvolging

Wie zit vast, waar, en hoe lang al. Automatisch opvolgen in plaats van maandelijks ontdekken.

Rapportering en synthese

Wekelijkse overzichten, board packs, statusrapporten. Data uit vijf systemen halen en er een leesbaar verhaal van maken.

Backoffice en administratie

Dossiers aanmaken, gegevens overzetten, statussen bijwerken, ontbrekende informatie opvragen.

Kennis ontsluiten

RAG over je eigen documentatie, contracten en historiek, zodat het team niet meer moet zoeken.

Drie weken ruis, en wat ik eruit leerde

Ik bouwde voor mezelf een agent die over al mijn systemen heen kijkt en me vertelt wat er aandacht nodig heeft. Office 365, CRM, GitHub, Azure DevOps, product analytics, Notion, facturatie. Technisch werkte hij vanaf dag één.

En drie weken lang vertelde hij me vooral dat er dingen gebeurden. Een mail binnengekomen. Een commit gepusht. Een deal verplaatst. Allemaal waar, allemaal nutteloos. Ik begon zijn berichten weg te klikken zonder ze te lezen, wat erger is dan geen agent hebben: het went.

Het probleem was niet het model en niet de integraties. Het was dat hij geen geheugen had van wat hij al gezegd had, en geen drempel voor wat de moeite waard was. Hij had geen idee wat normaal was, dus was alles nieuws.

- ✓ **Een signaal is pas een signaal als het afwijkt**

De agent moest eerst weten wat het normale patroon is voor hij iets kon melden.

- ✓ **Hij moet onthouden wat hij al gezegd heeft**

Een persistent register van open lussen, zodat dezelfde melding niet elke dag opnieuw komt.

- ✓ **Stilte is een geldige uitkomst**

Een dag zonder melding betekent dat er niets aan de hand is. Dat is precies wat je wil.

Dit is waarom ik geen agents bouw zonder eerst te meten wat er normaal gebeurt. Een agent die alles ziet en niets betekent, kost je meer aandacht dan hij bespaart.

Hoe ik een Agent Build aanpak

Eén proces, scherp afgebakend, vaste prijs die we vooraf afspreken. Van scope tot overdracht.

01

Scope vastzetten

Eén proces. We spreken vooraf af wat de agent mag beslissen, wat hij moet voorleggen, en wat hij nooit doet. Dat staat op papier voor er een regel code is.

02

Bouwen tegen de echte data

Geen synthetische testset. Ik bouw tegen je werkelijke systemen, in een omgeving waar fouten niets kapotmaken. Dat legt meteen bloot wat de briefing niet vertelde.

03

Guardrails en observeerbaarheid

Logging, foutafhandeling, escalatiepaden en een duidelijke drempel waarboven een mens beslist. Dit is precies het deel dat pilots overslaan.

04

Uitrol en overdracht

In productie, met documentatie en een team dat weet hoe het werkt en hoe je het aanpast. Ik ben geen afhankelijkheid.

Ik bouw dit ook voor mezelf

Ik ben Jarko Mesuere. Vandaag COO van een SaaS-scale-up, waar ik de volledige dagelijkse werking draag: commercieel, onboarding, finance en engineering delivery. Daarvoor vijftien jaar in IT: systeembeheer, infrastructuurlead op een nationaal tolplatform, engineering manager en program lead op de grootste parkeerapp van de Benelux.

Die volgorde is het punt. Ik heb Kubernetes-clusters beheerd, Terraform geschreven en releases teruggerold om drie uur 's nachts, en daarna geleerd hoe je een P&L stuurt en een board rapporteert. Dezelfde persoon die de vergadering leidt, schrijft de code. Dat scheelt een vertaalslag, en die vertaalslag is precies waar de meeste AI-projecten stranden.

Wat hierboven staat is geen theorie. Sentinel, mijn eigen monitoring-agent, leest over zeven systemen heen, houdt een persistent register bij van open lussen en classificeert elke voorgestelde actie in drie niveaus: zelf doen, voorleggen, of afblijven. Hij draait zelf gehost, achter Tailscale, want de data is gevoelig.

15 jaar

in IT en operations

7

stuurdashboards in één dag

4

structurele lekken blootgelegd

Drie manieren om te starten

Elk met een afgebakende scope en een vaste prijs die we afspreken voor je iets moet beslissen. Geen dagtarief dat oploopt.

Agent Build

Vaste prijs per proces

Eén concreet proces geautomatiseerd en in productie. Draaiend, met monitoring, foutafhandeling en eigenaarschap dat naar jouw team gaat.

- ✓ Procesontwerp en afbakening
- ✓ Guardrails, logging en escalatie
- ✓ Bouw, integratie en uitrol
- ✓ Overdracht en documentatie

Ops Scan

Vaste prijs, vooraf afgesproken

Weten waar het lekt voor je automatiseert. Een werkende cockpit op je échte cijfers plus een geprioriteerde lijst met wat een agent kan overnemen.

- ✓ Doorlichting van processen en systemen
- ✓ Geprioriteerde lijst met impactschatting
- ✓ Stuurcockpit op je eigen cijfers
- ✓ Voorstel voor wat eerst

Interim COO of program lead

Beperkte beschikbaarheid

Voor als er een gat zit op operations- of programmaniveau. Ik neem het over, breng het op orde en draag het over.

- ✓ Vast werkritme en governance
- ✓ Teamstructuur en eigenaarschap
- ✓ Sturing op cijfers die kloppen
- ✓ Overdracht als expliciet doel

Eén proces in gedachten?

Vertel me welk proces je het meest ergert. Ik zeg je in dertig minuten of het zich leent voor een agent en wat het ongeveer vraagt. Geen verkooppraat — als ik niet de juiste persoon ben, zeg ik dat, en meestal ken ik iemand die het wel is.

 +32 479 02 62 01

 jarko@jarko.be

 jarko.be

Ik antwoord meestal dezelfde dag.