

Vijf ontwerppatronen voor AI-agents, en hoe je ze in Dify bouwt

Bijlage bij de workshop AI Agents

LeX Consultancy B.V. · 20 september 2026



Wie “een agent” wil bouwen, moet eerst één vraag beantwoorden: hoeveel mag het model zelf beslissen? Het antwoord bepaalt hoe je bouwt, wat het kost en hoeveel er fout kan gaan. In het vak worden vijf patronen onderscheiden, van “helemaal niets beslissen” tot “beslissen, maar onder toezicht”. Je hebt ze in de handleidingen allemaal gebouwd zonder dat ze zo heetten.

De vuistregel: kies het eenvoudigste patroon dat het probleem oplost. Elk stapje omhoog kost meer tijd, meer credits en meer kans op een verrassing.

Patroon	Wie beslist de stappen	Voorbeeld uit de serie	Handleiding
1. Eén aanroep	Jij, vooraf; het model doet één ding	Tekst op niveau (één LLM-stap)	1
2. Redeneren en handelen (ReAct)	Het model, stap voor stap, met gereedschap	Schoolgids-assistent, Ouderbrief-assistent (agent met kennisbank)	2, 3
3. Plannen en uitvoeren	Eerst een plan, dan losse uitvoerders	Schoolnieuws (drie kanalen uit één boodschap), Toetsweekplanner	3, 4
4. Zelfkritiek	Het model beoordeelt en verbetert zijn eigen werk	De Controleur in Schoolnieuws met controle; de Lesmateriaal-check	7, 6
5. Poortwachter	Een onafhankelijke controle laat het resultaat wel of niet door	De Poortwachter (code) in Schoolnieuws met controle; de mens in de lus van de Studiecoach	7, 8

1. Eén aanroep (single-shot)

Hoe het werkt. De gebruiker levert iets aan, het model doet precies één bewerking, klaar. Geen lus, geen gereedschap, geen keuzes onderweg.

invoer → [LLM: één instructie] → uitvoer

Kies dit als de taak elke keer hetzelfde is: classificeren, samenvatten, herschrijven, iets uit een tekst halen, een formulier omzetten naar een vaste indeling.

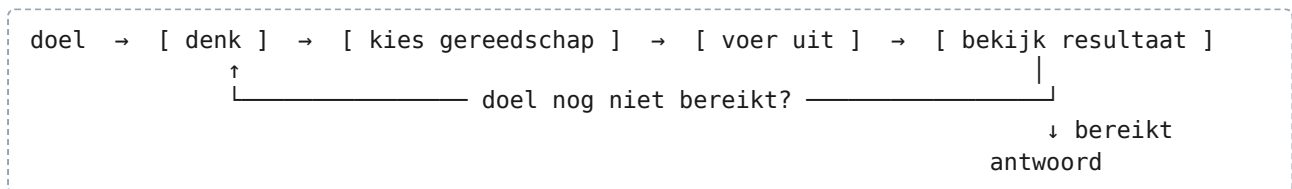
Onderwijsvoorbeeld. “Zet deze ouderklacht om in een gestructureerd formulier: onderwerp, klas, urgentie, gevraagde actie.”

In Dify. Een Workflow met User Input → LLM → Output. Twee LLM-stappen achter elkaar (zoals in handleiding 1: herschrijven en dan vragen maken) is nog steeds dit patroon: jij bepaalt de volgorde, het model beslist niets.

Voordelen. Snel, goedkoop, voorspelbaar, makkelijk te testen. **Nadelen.** Kan niet omgaan met onduidelijke invoer en kan geen informatie opzoeken. Als de gebruiker iets vergeet, komt er een verkeerd antwoord in plaats van een vraag.

2. Redeneren en handelen (ReAct)

Hoe het werkt. Het model krijgt een doel en gereedschap. Het denkt na over de volgende stap, kiest een gereedschap (zoeken, rekenen, een kennisbank raadplegen, een API aanroepen), bekijkt het resultaat en beslist opnieuw. Dat herhaalt zich tot het doel is bereikt.



Kies dit als het model iets moet opzoeken of uitproberen en je vooraf niet weet hoeveel stappen dat kost: vragen beantwoorden uit documenten, webonderzoek, iets uit een database halen, foutzoeken.

Onderwijsvoorbeeld. “Wat zijn de regels voor verlof buiten de vakanties, en wat moet ik als mentor dan doen?” De agent zoekt in de schoolgids, vindt twee hoofdstukken, combineert ze.

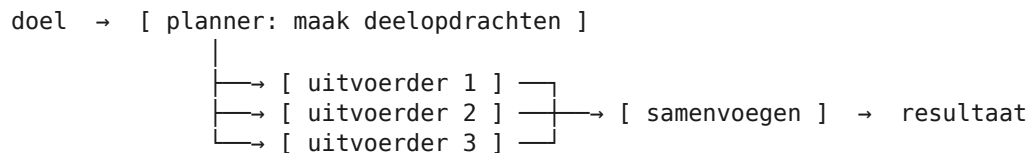
In Dify. De Agent-app in Studio (handleiding 2 en 3): instructie plus Knowledge plus Tools; de agent beslist zelf wanneer hij zoekt. En de Agent Console (handleiding 4), waar het model ook bestanden kan lezen, code kan draaien en op het web kan zoeken. Het label “Used Knowledge” boven een antwoord is dit patroon in het klein: het model besloot te zoeken.

Voordelen. Kan met onverwachte vragen omgaan, gebruikt echte informatie, is flexibel.

Nadelen. Trager en duurder (elke lus is een modelaanroep), meer kans op een zijpad, en je hebt grenzen nodig: maximaal aantal stappen, welke gereedschappen wel en niet, wat te doen bij een fout. In Dify zijn dat de regels in de instructie en de keuze welke Tools je aanzet.

3. Plannen en uitvoeren (planner-executor)

Hoe het werkt. Eén stap maakt eerst een plan met deelopdrachten. Daarna voert per deelopdracht een aparte stap (of een aparte agent) het werk uit, waar mogelijk naast elkaar. Aan het eind worden de resultaten samengevoegd.



Kies dit als de taak uit herkenbare delen bestaat die los van elkaar kunnen: een rapport met vaste hoofdstukken, een bericht voor meerdere kanalen, een analyse in stappen (verzamelen, analyseren, schrijven, controleren).

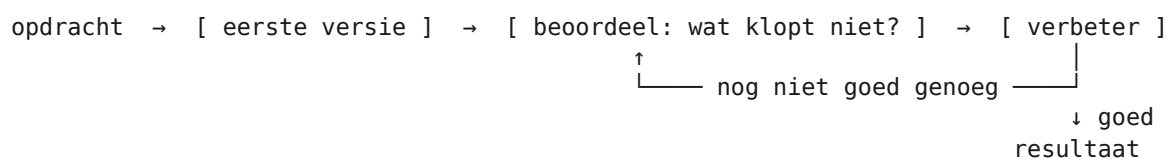
Onderwijsvoorbeeld. Eén kernboodschap wordt een websitebericht, een ouderappbericht en een socialbericht, elk volgens de eigen regels van dat kanaal.

In Dify. Schoolnieuws (handleiding 3) is dit patroon met een vast plan: jij schreef het plan als drie LLM-blokken met eigen prompts, plus een Output die samenvoegt. Wil je dat het model zelf plant, dan zet je een eerste LLM-blok in dat een lijst deelopdrachten maakt, en laat je een **Iteration**-node die lijst afwerken. In de Agent Console doet het model dit uit zichzelf bij grotere opdrachten: het schrijft eerst een plan en werkt dat af (de Toetsweekplanner in handleiding 4 laat dat zien).

Voordelen. Complexe taken worden overzichtelijk, elke uitvoerder is apart te testen en te verbeteren, en delen kunnen parallel. **Nadelen.** Meer bouwwerk, en een slecht plan levert netjes uitgevoerd verkeerd werk op. Laat het plan zichtbaar zijn, zodat een mens het kan corrigeren voordat de uitvoerders starten.

4. Zelfkritiek (reflexive)

Hoe het werkt. Het model maakt een eerste versie, beoordeelt die daarna zelf aan de hand van criteria, en verbetert. Eventueel een paar rondes, tot de kwaliteit goed genoeg is.



Kies dit als kwaliteit belangrijker is dan snelheid en de criteria benoembaar zijn: teksten voor publicatie, code, lesmateriaal, alles waar “goed genoeg” een lijstje eisen heeft.

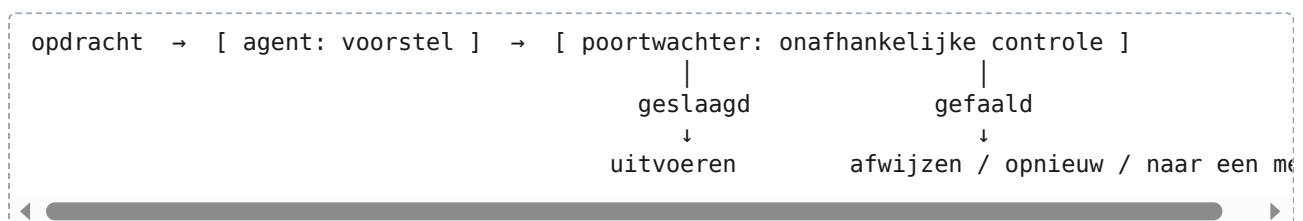
Onderwijsvoorbeeld. “Schrijf een leerlingversie van deze tekst op B1-niveau. Controleer daarna zelf: zinnen langer dan vijftien woorden? Woorden uit de lijst te vermijden woorden? Ontbrekende stap in de uitleg? Herschrijf waar nodig.”

In Dify. Een Workflow met twee LLM-blokken: het eerste schrijft, het tweede krijgt de tekst plus de criteria en de opdracht “beoordeel en verbeter”. Wil je meerdere rondes, dan gebruik je een **Loop**-node met een stopvoorwaarde. Goedkoper alternatief dat vaak al genoeg is: één prompt met de opdracht om eerst te schrijven, dan te controleren op de genoemde punten, en alleen de verbeterde versie te geven. Handleiding 6 (lesmateriaal aanpassen) is de plek waar dit patroon thuishoort.

Voordelen. Merkbaar betere teksten, minder slordigheden, en de criteria dwingen je om te benoemen wat “goed” is. **Nadelen.** Twee tot drie keer de kosten en de wachttijd. En let op: het model beoordeelt zichzelf. Een feitelijke fout die het zelf gelooft, blijft staan. Zelfkritiek verbetert vorm en volledigheid, geen waarheid.

5. Poortwachter (verifier-gated)

Hoe het werkt. Het resultaat van de agent gaat niet direct naar buiten, maar langs een onafhankelijke controle: een vaste regelset, een schema, een berekening, een tweede model met alleen de controletaak, of een mens. Alleen wat door de poort komt, wordt uitgevoerd of getoond; wat faalt, wordt afgewezen, opnieuw geprobeerd of doorgestuurd naar een mens.



Kies dit als een fout echt schade doet: geld, cijfers, communicatie naar ouders, alles met persoonsgegevens, alles wat onder beleid of wetgeving valt.

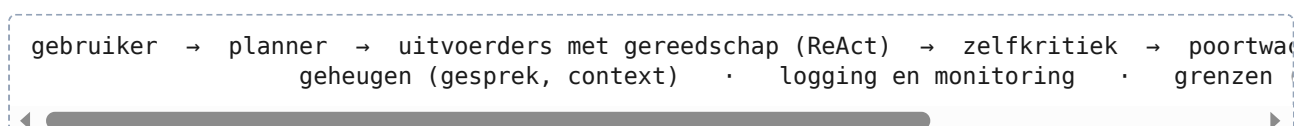
Onderwijsvoorbeeld. Een agent stelt een bericht voor de ouderapp op. De poortwachter controleert: geen namen van leerlingen, geen woorden uit de verboden lijst, maximaal 80 woorden, een datum die in de toekomst ligt. Faalt één regel, dan gaat het bericht naar de communicatiemedewerker met de reden erbij.

In Dify. Een **Code**-node of een **IF/ELSE**-node na de LLM: harde regels (lengte, verboden woorden, aanwezige velden) controleer je met code, niet met een model. Voor zachtere regels een tweede LLM-blok met uitsluitend de vraag “voldoet dit aan de volgende regels, antwoord met JA of NEE en de reden”. De uitkomst stuurt de IF/ELSE: door naar Output, of naar een tak die het bericht met de reden aan een mens voorlegt. Een **Human Input**-node is de meest eerlijke poortwachter die er is.

Voordelen. Voorspelbaar en uitlegbaar: je kunt precies zeggen waarom iets wel of niet doorging. Verplichte regels worden ook echt afgedwongen, niet alleen “gevraagd” in een prompt. **Nadelen.** Extra stap, en de regels moeten goed zijn: een te strenge poort houdt alles tegen, een te losse doet niets. Laat de agent nooit zijn eigen poortwachter zijn.

Combineren

In de praktijk stapel je patronen. Een volwassen toepassing ziet er vaak zo uit:



Voor de school betekent dat:

- **Begin bij patroon 1.** Als een vaste workflow werkt, is een agent overbodig. De meeste taken op een school zijn patroon 1 of 3.
- **Ga naar patroon 2 als de invoer wisselt** en er gezocht moet worden. Zet dan meteen grenzen: welke bronnen, hoeveel stappen, wat bij twijfel.
- **Voeg 4 toe als de uitvoer gepubliceerd wordt** en er criteria zijn.
- **Voeg 5 toe zodra er iets naar buiten gaat** of er persoonsgegevens in het spel zijn. Een mens in de lus is een geldige, vaak de beste, poortwachter.
- **Log alles.** In Dify staat elke run onder **Logs**. Zonder logboek kun je niet leren van fouten en niet uitleggen wat er gebeurde.

Voor handleiding 6. De lesmateriaal-agent gebruikt alle vijf: een vaste intake (1), zoeken naar open leermateriaal (2), een plan per leerlingprofiel (3), zelfcontrole op taalniveau en volledigheid (4), en een poort voor licenties en persoonsgegevens voordat er iets wordt opgeleverd (5).

Om te onthouden

- Het patroon bepaalt hoeveel het model zelf beslist. Kies het laagste dat werkt.
- Jij hebt patroon 1 (handleiding 1), 2 (handleiding 2, 3, 4, 6), 3 (Schoolnieuws, Toetsweekplanner), 4 (de Controleur) en 5 (de Poortwachter, de mens in de lus) gebouwd.
- Zelfkritiek verbetert vorm, geen waarheid. Een poortwachter dwingt af, een prompt vraagt alleen.
- Geen enkel patroon garandeert een goed antwoord. Ontwerp voor de fout: wat gebeurt er als het misgaat, en wie ziet dat?

Licentie: CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/deed.nl>). Je mag dit document kopiëren, bewerken en hergebruiken, ook op je eigen school, als je de bron noemt (Guido van Dijk, LeX Consultancy B.V. en AI Netwerk Limburg, <https://git.lexconsultancy.nl/guido/dify-handleidingen>) en je bewerking onder dezelfde licentie deelt. Het logo van AI Netwerk Limburg valt buiten de licentie; namen van producten zijn merken van hun eigenaren. Scholen, personen en documenten in de voorbeelden zijn verzonnen.

Eigenaar en contact: Guido van Dijk, LeX Consultancy B.V. en AI Netwerk Limburg · guido@l-e-x.nl · guido@ai-netwerk-limburg.nl



AI Netwerk Limburg · LeX Consultancy B.V. · Vijf ontwerppatronen voor AI-agents, en hoe je ze in Dify bouwt