limescape | avans hogeschool

HAND-OUT EN LESOVERZICHT

Je kreeg wat je promptte, maar niet wat je bedoelde.

Waarom professionele AI-oplossingen niet zonder evaluaties kunnen.

Hand-out bij het gastcollege voor de module AI van Avans Hogeschool, 19 november 2026. Door Rodger Blom, AI engineer bij TrueLime.

Een AI-oplossing die in een demo goed werkt, is daarmee nog niet goed genoeg voor de praktijk. Een demo laat zien dat het goed kán gaan. Met evaluaties toets je doorlopend hoe de hele oplossing presteert, zodat je de kwaliteit van de uitvoer kunt bewaken. In het college hielden we het eenvoudig, met één agent en één prompt. Deze hand-out vat het college samen en laat zien hoe je dezelfde werkwijze toepast op complexere oplossingen uit de praktijk.

Lesoverzicht

Het college duurde 90 minuten en bestond uit vijf delen.

DEEL	ONDERWERP
1	Opening: een prompt is geen specificatie
2	Hoe snel AI-modellen veranderen en welke risico's dat geeft
3	Observability, evaluaties en de drie kloven tussen idee en productie
4	Het spel: in vier rondes beoordelen of een AI-agent live mag
5	Afsluiting: wat neem je mee?

1. Een prompt is geen specificatie

Het college begon met een fragment uit de film Terminator 2 uit 1991. De terminator krijgt de opdracht: "Je mag niemand doden." Hij houdt zich daaraan, maar schiet daarna wel een bewaker in zijn benen. De zaal stemde over de vraag of de terminator "de prompt" had uitgevoerd.

Hij heeft niemand gedood, maar dit was niet de bedoeling. De scène is 35 jaar oud en toch gaat het met AI-agents vandaag op dezelfde manier mis: een agent doet wat er staat, niet wat je bedoelde. Een prompt is geen specificatie, en een geslaagde demo is geen evaluatie. De vraag van het college was daarom: hoe toon je aan dat een AI-oplossing doet wat je bedoelt?

2. AI-modellen veranderen steeds sneller

In november 2026 lieten onderzoekers van MIT het model GPT-6 Astra de gewichten van een ander, veel kleiner model rechtstreeks aanpassen. De score van dat kleine model op een wiskundetoets steeg daardoor van 7,9% naar 48,1%. Een krantenkop maakte daarvan: "AI herschrijft nu zijn eigen brein". Die kop klopt maar voor een deel, want het model paste een ander model aan en niet zichzelf. Ook een krantenkop moet je dus evalueren: wat is er precies gemeten, en door wie?

Ook de risico's nemen toe. Volgens Reuters gebruikte één hacker in oktober 2026 samenwerkende AI-agents om ruim 1.200 websites in vier landen aan te vallen. Er is ook een tegengeluid: volgens Axios leverden miljoenen pogingen weinig geslaagde aanvallen op.

Voor jouw werk betekent dit dat het model waarop je oplossing draait steeds vaker en steeds sneller verandert. Wat vorige maand goed werkte, kan na een nieuwe versie anders uitpakken. Eén keer testen bij de oplevering is daarom niet genoeg.

3. De praktijk is complexer dan één prompt

In het college stond één prompt centraal, om het eenvoudig te houden. Een AI-oplossing in de praktijk bestaat uit veel meer onderdelen. Er zijn twee hoofdvormen, die vaak worden gecombineerd.

- **Deterministische workflows met AI-componenten.** De stappen en hun volgorde liggen vast. In een deel van de stappen doet een AI-model het werk, bijvoorbeeld gegevens uit een document halen of een samenvatting schrijven.
- **Autonome AI-agents.** De agent bepaalt zelf welke stappen hij zet om een doel te bereiken. Hij werkt met prompts, roept tools aan (tool calls), gebruikt skills en zet subagents in voor deeltaken.

In zo'n oplossing kan een fout op veel plaatsen ontstaan: in een bron, in een prompt, in de keuze voor een tool, in wat een subagent teruggeeft of in een gewone stap vóór of na het model. Aan het eindresultaat zie je vaak niet waar het misging. Een agent kiest bovendien niet elke keer dezelfde weg, dus dezelfde vraag kan de volgende keer via andere stappen worden beantwoord.

Dat bleek ook uit de casus die we in het college bekeken. Bij een AI-oplossing die dossiers analyseert, stonden alle automatische controles op groen en had een mens de uitkomst goedgekeurd. Toch zat er een fout in de analyse. De oorzaak lag in een stap vóór het taalmodel. Een tweede AI-model vond de fout door elke bewering met de bron te vergelijken.

4. Observability en evaluaties

Om de kwaliteit van zo'n oplossing te bewaken, heb je twee dingen nodig.

Observability (meekijken). Je wilt kunnen nagaan wat er binnen de oplossing gebeurt. Daarvoor leg je bij elke vraag vast wat het systeem binnenkrijgt, welke stappen het uitvoert, welke tools en subagents het aanroept en welk antwoord het geeft. Zo'n vastlegging heet een trace. Met traces kun je nagaan in welke stap een fout is ontstaan.

Evaluaties (evals). Het doel van evaluaties is doorlopend te toetsen hoe de hele oplossing presteert, zodat je de kwaliteit van de uitvoer kunt bewaken. Je laat een vaste set vragen of taken steeds opnieuw uitvoeren en beoordeelt de uitkomsten. Daarbij kijk je naar drie dingen.

- **Het eindresultaat.** Is de uitvoer juist en bruikbaar voor de gebruiker?
- **De afzonderlijke onderdelen.** Koos de agent de juiste tool, gebruikte hij de juiste bron en gaf een subagent door wat nodig was?
- **Het verloop in de tijd.** Blijft de kwaliteit gelijk na een wijziging in het model, een prompt, een tool, een skill of een bron? Het is ook belangrijk om je evaluaties actueel te houden ná de livegang in productie. Vul ze steeds aan met belangrijke cases die je dan tegenkomt. Zo worden je evaluaties steeds betrouwbaarder.

Evaluaties geven geen zekerheid. Een set vragen dekt nooit alles wat er in productie binnenkomt. Daarom vul je de set aan met praktijkgevallen en combineer je codecontroles, beoordelingen door een tweede AI-model en het oordeel van mensen.

Drie kloven tussen idee en productie

Shreya Shankar en Hamel Husain beschrijven drie kloven (in het Engels: *gulfs*) tussen een idee en een oplossing die in productie goed werkt. Evaluaties helpen je die kloven te overbruggen.

1. **Begrijpen.** Je moet het proces en de echte data door en door kennen. Dat lukt het best door echte voorbeelden en traces te bekijken.
2. **Specificeren.** Je vertaalt de use case of het proces van een organisatie naar prompts, agents, tools en skills. Wat je niet opschrijft, vult het model vaak zelf in. Dan krijg je wat je promptte, maar niet wat je bedoelde.
3. **Generaliseren.** Dit is de moeilijkste kloof. Ook met een goede prompt gaat het model bij bepaalde invoer de mist in, zeker bij edge cases waaraan je niet had gedacht.

5. Het spel: een vereenvoudigd voorbeeld

Om de werkwijze te oefenen, gebruikten we in het college bewust een eenvoudig voorbeeld: één AI-agent voor de klantenservice van een schoenenwebshop, met één prompt en twee bronnen. In vier rondes beoordeelden we of deze agent live mag.

- **Ronde 1: de demo.** De zaal zag drie goede antwoorden en stemde over de vraag of de agent live mag. Daarna zag de zaal twee foute antwoorden van dezelfde agent, van dezelfde dag. Een paar goede antwoorden zeggen dus weinig over de rest.
- **Ronde 2: kijk naar de data.** Iedereen beoordeelde zelf vier antwoorden, met de bronnen ernaast. De zaal was het niet over elk antwoord eens. Daarom leg je vooraf in beoordelingsregels vast wat goed en fout is.

- **Ronde 3: bouw de controle.** Iedereen schreef een instructie voor een LLM as a judge: een taalmodel dat de antwoorden van de agent beoordeelt. De app vergeleek het oordeel van elke judge met dat van de mens. In de traces zagen we daarna wat het model per aanroep binnenkreeg en teruggaf.
- **Ronde 4: de verbeterde versie.** De prompt was aangepast en bijna alle gevonden fouten waren aangepakt. Toch ging het mis bij een vraag waarop de oude versie nog een goed antwoord gaf.

Bij een workflow of een autonome agent werk je op dezelfde manier. Je beoordeelt dan ook de tussenstappen in de traces en niet alleen het eindresultaat.

Een LLM as a judge zet je pas als laatste in

Een judge gebruik je om grote aantallen uitkomsten te controleren, maar pas nadat je zelf hebt geleerd die uitkomsten te beoordelen. Daar gaan vier stappen aan vooraf.

1. **Lees de uitkomsten.** Beoordeel zelf tientallen uitkomsten van je oplossing, met de bronnen en de traces ernaast.
2. **Groepeer de fouten.** Kijk welke soorten fouten vaak voorkomen en in welk onderdeel de oorzaak zit.
3. **Los op wat je kunt.** Een fout in een prompt, een tool of een bron los je eerst op.
4. **Controleer met code.** Gebruik code voor controles die je met een eenvoudige regel kunt uitvoeren, bijvoorbeeld op bedragen of datums.
5. **Zet een judge in.** Toets eerst of de judge tot hetzelfde oordeel komt als de mens. Kijk daarbij ook hoeveel fouten de judge vindt en hoeveel goede uitkomsten hij onterecht afkeurt.

6. Wat neem je mee?

1. **Een demo is geen evaluatie.** Een paar goede uitkomsten zeggen weinig over hoe de oplossing zich bij andere invoer gedraagt. Leg een vaste set vragen of taken vast en beschrijf wat een goede uitkomst is. Stel die set samen met de mensen uit de praktijk.
2. **Kijk eerst zelf naar de uitkomsten en de traces.** Groepeer de fouten en zoek per foutsoort uit waar de oorzaak zit. Pas daarna automatiseer je de controle en meet je of die controle klopt.
3. **Evalueren doe je doorlopend.** Toets de hele oplossing opnieuw als je het model, een prompt, een tool, een skill of een bron verandert, en blijf ook in productie meten. Zo zie je op tijd dat de kwaliteit terugloopt.

Meer weten?

Stuur me een berichtje op LinkedIn via **linkedin.com/in/rodgerblom** of neem contact met ons op via **limescape.ai**.

Rodger Blom, AI engineer bij TrueLime

Bijlage: begrippen en bronnen

Begrippen

BEGRIP	BETEKENIS
Prompt	De invoer, meestal tekst, waarmee je een AI-model stuurt. Niet alleen taalmodellen werken met prompts, maar ook modellen voor beeld en spraak.
Workflow	Een vaste, herhaalbare reeks stappen om een taak uit te voeren. In een workflow met AI-componenten voert een AI-model een of meer van die stappen uit. De volgorde van de stappen ligt vast.
AI-agent	Een AI-systeem dat zelfstandig beslissingen neemt en acties uitvoert om een doel te bereiken. Bij een agent op basis van een taalmodel bepaalt het model zelf de stappen en de tools die het gebruikt.
Subagent	Een agent die een deeltaak van een andere agent krijgt, deze met eigen instructies en tools uitvoert en het resultaat teruggeeft.
Tool en tool call	Een tool is een functie die een model mag gebruiken, bijvoorbeeld om in een bron te zoeken of iets te berekenen. Met een tool call vraagt het model om die functie uit te voeren. De applicatie voert de functie uit en geeft het resultaat terug aan het model.
Skill	Een pakket met instructies en eventueel scripts en andere bestanden, dat een agent pas laadt als een taak erom vraagt.
Observability	Het vermogen om van buitenaf te begrijpen wat er binnen een systeem gebeurt, aan de hand van wat het systeem vastlegt: logs, meetwaarden en traces.
Trace	De vastlegging van de weg die één vraag of taak door het systeem aflegt. Een trace bestaat uit spans: de afzonderlijke stappen, zoals een aanroep van een model of een tool.
Evaluatie (eval)	Een toets van de uitvoer van een model of een AI-oplossing aan criteria die je vooraf vastlegt. Het doel is doorlopend de kwaliteit van de uitvoer te bewaken.
Beoordelingsregels	De vooraf vastgelegde criteria waaraan je een uitkomst toetst.
LLM as a judge	Een taalmodel dat de uitvoer van een model of een AI-oplossing beoordeelt aan de hand van een instructie met criteria. Je toetst de judge eerst aan de oordelen van een mens met kennis van het onderwerp.
Edge case	Een situatie die alleen optreedt bij een uiterste invoerwaarde of onder extreme omstandigheden. Een edge case kan voorzien of onvoorzien zijn.
Gewichten	Getallen in een model waarmee het zijn invoer vermenigvuldigt. Ze worden tijdens de training bepaald.

Bronnen

- The Register en R&D World, 3 november 2026, over het onderzoek van MIT waarin een model de gewichten van een ander model aanpaste.
- Reuters, 19 oktober 2026, over de aanval met AI-agents op ruim 1.200 websites.
- Axios, 28 oktober 2026, over het geringe aantal geslaagde aanvallen.
- Shreya Shankar en Hamel Husain, *AI Evals for Engineers & PMs*. In deze cursus beschrijven zij de drie kloven tussen idee en productie.
- De begrippen zijn getoetst aan Wikipedia, de documentatie van Anthropic, OpenAI en OpenTelemetry, de Machine Learning Glossary van Google en de gids *Using LLM-as-a-Judge For Evaluation* van Hamel Husain.