Inhoud

Inleiding

AI-agents zijn in korte tijd van demo naar werkvloer gegaan. In veel organisaties zie je dezelfde beweging: een eerste experiment met een taalmodel levert verrassend bruikbare antwoorden op, daarna volgt al snel de wens om het niet bij praten te laten maar ook te laten doen. Een agent die tickets beoordeelt, een agent die een dossier voorbereidt, een agent die gegevens opzoekt en een voorstel maakt, een agent die taken in een workflow afhandelt. Op dat moment verandert het karakter van het werk. Je bouwt niet langer een slimme chat, je bouwt een stuk operationele software dat met echte processen, echte systemen en echte risico's verweven raakt.

Dit boek vertrekt vanuit één eenvoudige observatie: het moeilijkste aan agents is meestal niet het eerste prototype, maar de stap naar een voorspelbare toepassing in productie. In een demo mag een antwoord soms nét niet kloppen, in productie is dat een incident. In een demo is een time-out vervelend, in productie breekt het je doorlooptijd, je kostenplafond en je vertrouwen. In een demo is het oké dat niemand precies weet waarom de agent iets deed, in productie moet je het kunnen uitleggen, reproduceren en waar nodig terugdraaien. Het centrale thema is daarom niet "hoe maak ik een agent slim", maar "hoe maak ik agentgedrag beheersbaar".

Filosofie achter de opzet

De opzet van dit boek is bewust anders dan veel introducties tot generatieve AI. Je vindt hier geen verzameling losse prompttrucs, geen lijst met tools die je 'moet kennen' en geen route die begint bij technologie en eindigt bij een vaag businessvoordeel. De ruggengraat is een praktische levenscyclus die je van idee naar inzet brengt, met expliciete keuzes, documentatie en overnamecriteria per stap.

Daarmee volgen we een principe dat je door het hele boek terugziet: maak het probabilistische deel zo klein mogelijk en het deterministische deel zo groot mogelijk. Het taalmodel is statistisch en kan variëren. Alles eromheen kun je ontwerpen, begrenzen, meten en testen. Dat betekent dat je vroeg in het traject afspreekt wat goed is, wat genoeg is en wanneer je stopt. Het betekent ook dat je niet probeert om alles met één magische prompt op te lossen, maar dat je bouwt met bouwstenen: planning, tools, geheugen, evaluatie, guardrails, interfaces, observability en operations.

De structuur is daarom een routekaart. Je begint bij waarde en afbakening, gaat via ontwerpkeuzes naar concrete bouwstenen, werkt daarna door naar betrouwbaarheid, evaluatie, observability, veiligheid en tot slot kosten, performance en gebruikerservaring. In elk hoofdstuk zit dezelfde ondertoon: je neemt een agent serieus als onderdeel van je systeemlandschap. Dat vraagt om dezelfde discipline die je al kent van integraties, security, operations en kwaliteitsborging, maar met een nieuw element: modelgedrag dat nooit volledig deterministisch wordt.

Centraal in het boek: het Agent Design Canvas

Veel agentprojecten lopen vast doordat teams te vroeg discussiëren over tools, modellen of frameworks. Dan krijg je gesprekken die technisch lijken, maar inhoudelijk onbeslist zijn: wat is de scope, wat is de waarde, welke data mag je gebruiken, welke fouten zijn acceptabel, wanneer moet een mens ingrijpen, wat is het kostenplafond enzovoort. Als die vragen niet expliciet beantwoord zijn, schuiven ze door naar later en komen ze terug als vertraging, herwerk of risico.

Daarom start dit boek met het Agent Design Canvas. Het canvas dwingt je om de casus scherp te krijgen, de *value metric* te kiezen en de risico's te benoemen voordat je gaat ontwerpen. Het vormt een gedeeld referentiepunt voor proceseigenaar, architectuur, security, privacy en oplevering. Belangrijker nog: het canvas is geen eenmalig formulier, maar een levend document. Zodra je scope

wijzigt, een extra tool aansluit, een strengere guardrail nodig hebt of kosten anders uitpakken, werk je bij en leg je vast waarom.

Het canvas maakt ook iets anders mogelijk: het legt de lat voor evaluatie en operatie vast. Je definieert niet alleen wat de agent moet doen, maar ook hoe je gaat meten of hij dat goed doet en hoe je ingrijpt als het gedrag afwijkt. Daarmee voorkom je dat kwaliteit een mening wordt of dat je in productie pas ontdekt dat je geen zicht hebt op wat er gebeurt.

Voor wie dit boek is bedoeld?

Dit boek is geschreven voor teams die agents willen inzetten in echte processen. Het is expliciet gericht op organisaties die met bestaande systemen werken, met nalevingseisen te maken hebben en die hun oplossingen moeten beheren. Je hoeft geen onderzoeker te zijn of een specialist in modeltraining. Je moet wel bereid zijn om agentontwikkeling te behandelen als productontwikkeling: met afspraken, versies, tests, monitoring en verantwoordelijkheden.

Concreet is dit boek bruikbaar voor:

- proces- en producteigenaren die casussen kiezen, waarde meten en scope bewaken;
- architecten en engineers die agentpatronen, integraties en runtime-architectuur ontwerpen;
- verantwoordelijken voor security, privacy en naleving die guardrails, datastromen en governance toetsen;
- verantwoordelijken voor operations en SRE (*site reliability engineering*) die productiegedrag moeten kunnen volgen, herstellen en auditen;
- verantwoordelijken voor UX (*user experience*) en servicedesign die agentinteractie begrijpelijk en betrouwbaar moeten maken.

De toon is daarom zakelijk en praktisch. We leggen uit wat je moet beslissen en welke documenten je nodig hebt. We kiezen waar mogelijk voor patronen en checklists in plaats van uitgebreide frameworks. Als je een hoofdstuk afrondt, heb je iets in handen dat je in een werksessie kunt gebruiken: een canvas, een specificatie, een rubric, een testset, een dashboardvoorstel of een draaiboek.

Wat je wel krijgt en wat niet

Je krijgt een end-to-endaanpak voor agents in productie, inclusief:

- patroonkeuzes voor agentgedrag en verwerkingsketens;
- toolintegraties als volwaardige API's met contracten, idempotentie en foutstrategieën;
- geheugen en RAG als beheersbare component met bronbeheer en bewaartermijnen;
- evaluatie en testen met golden sets, rubrics en CI-geschikte regressie;
- observability en operations met logging, tracing, SLI's, SLO's, alerts en draaiboeken;
- veiligheid, privacy en naleving bekeken door een Europese bril;
- kosten en performance met tokenbudgettering, caching en modelcascades;
- UX en servicedesign voor conversatiepatronen en mens-in-de-lusprocessen.

Wat je niet krijgt is een boek vol kant-en-klare code in één taal of één platform. Dat is een bewuste keuze. Tooling verandert snel, frameworks hebben hun eigen opinies en voorbeeldcode zonder testmogelijkheid geeft schijnzekerheid. In plaats daarvan beschrijven we patronen die je in elke moderne stack kunt implementeren. Waar code normaal een 'bewijs' is, zijn in dit boek contracten, schema's, tests en meetpunten het bewijs. Daarmee blijf je onafhankelijk van leveranciers en houd je controle over kwaliteit.

Diezelfde keuze zie je terug in de appendix over toolservices: geen framework-discussie, wel een helder request- en responsemodel, endpointpatronen, authenticatie-uitgangspunten, observability-eisen en testvormen. Dit is precies de laag waar veel agentprojecten winnen of verliezen. Niet omdat het ingewikkeld is, maar omdat het vaak te impliciet blijft.

Hoe je dit boek gebruikt

Je kunt het boek lineair lezen, maar het is ook ontworpen als werkboek. Elk hoofdstuk heeft een eigen doel en levert materiaal op dat je direct kunt gebruiken. Er zijn twee effectieve manieren om ermee te werken.

De eerste is de projectroute. Je volgt hoofdstuk 1 tot en met 4 om van idee naar ontwerpkeuzes te gaan. Daarna werk je hoofdstuk 5 tot en met 8 uit om promptgedrag, geheugen, tools en evaluatie productierijp te maken. Vervolgens behandel je hoofdstuk 9 tot en met 11 om observability, veiligheid en kosten te borgen. Hoofdstuk 12 helpt je om de agent goed te laten landen bij gebruikers

en in serviceprocessen. Dit is de route voor teams die daadwerkelijk willen bouwen en uitrollen.

De tweede manier is rolgericht lezen. Een proceseigenaar kan starten bij hoofdstuk 1 en daarna selectief lezen in hoofdstuk 9, 10 en 11 om afspraken over operatie, naleving en budget te begrijpen. Een architect kan hoofdstuk 3 en 4 als basis nemen en daarna verdieping zoeken in toolintegraties, geheugen en patronen voor betrouwbaarheid. Operations kan juist beginnen bij observability en draaiboeken en terugbladeren naar ontwerpkeuzes die daar consequenties voor hebben. De hoofdstukken zijn zo geschreven dat je kunt instappen zonder eerst alles te lezen, zolang je het canvas en de kernbegrippen kent.

Positionering in het Europese kader

Agents raken al snel persoonsgegevens, besluitvorming en procesverantwoordelijkheid. Daarom is het boek nadrukkelijk EU-gericht. De AVG noch de AI-verordening zijn bijzaken, maar een ontwerpbeperking. Hetzelfde geldt voor governance rond rollen en rechten, datalocatie, bewaartermijnen, audittrail en aantoonbaarheid. Dit boek probeert niet om juridische handboeken te vervangen. Het biedt een praktische denklijn waarmee je als team de juiste vragen stelt, maatregelen kiest en bewijs verzamelt.

Dat zie je ook terug in de structuur: privacy en naleving worden niet los behandeld, maar gekoppeld aan concrete bouwstenen. Dataminimalisatie raakt logging en geheugen. Rolgebaseerde toegang raakt tooladapters. Bewaartermijnen raken vectorstores, logs en evaluatiesets. Auditability is geen ‘rapportje achteraf’, maar een eigenschap die je ontwerpbaar maakt via correlatie-id’s, versie-informatie en traceerbare beslissingen.

Uitgebreid en compleet

In veel boeken over AI ligt de nadruk op mogelijkheden. In dit boek ligt de nadruk op grenzen. Dat klinkt zwaarder, maar het is in de praktijk juist bevrijdend. Als je vooraf weet wat de agent wel en niet mag doen, hoe je fouten afhandelt, hoe je kosten begrenst en wanneer een mens ingrijpt, kun je sneller bouwen met minder discussie. Compleetheid is hier geen doel op zich, maar een manier om herwerk te voorkomen.

Daarom eindigen hoofdstukken met samenvattingen en overnamecriteria. Ze maken expliciet wanneer je door kunt naar de volgende fase. Niet omdat je dan ‘klaar’ bent, maar omdat je genoeg houvast hebt om de volgende stap verant-

woord te zetten. Dat past bij hoe agentprojecten in echte organisaties lopen: iteratief, met stakeholders, met audits, met beheerlast en met veranderende omstandigheden.

Nog meer weten? Gebruik de chatbot!

Bij dit boek zit een gratis chatbot die getraind is met onder andere de inhoud van dit boek. De chatbot beschikt daarnaast over meer informatie over (het bouwen van) agents en kan ook concrete cases voorstellen. Je krijgt gratis toegang als je dit boek registreert (zie de informatie op de pagina *Lezersvoordeel* aan het begin van dit boek).

1 Van Idee naar Casus

Introductie en leesdoel

Dit hoofdstuk helpt je om een ruw idee te vertalen naar een afgebakende casus met een meetbare uitkomst. Je werkt met het Agent Design Canvas, een eenduidig invulraamwerk waarin je het probleem, de doelgroep, de scope, de waarde en de risico's vastlegt. Het doel is niet om al te ontwerpen of te bouwen, maar om scherp te krijgen wat de agent moet opleveren, onder welke voorwaarden en wanneer je stopt.

Relevantie

Zonder gedeeld begrip van probleem en waarde verzand je snel in discussies over tools of modelkeuzes. Het canvas voorkomt dat. Je bepaalt eerst waarom een agent nodig is, wie erdoor geraakt wordt en hoe je succes objectief meet. Dat levert tijdwinst op in de realisatie, verlaagt de kans op scope-kruip en maakt beter inzichtelijk waar privacy- en nalevingsrisico's liggen. Het canvas is bovendien het document waarop stakeholders besluiten of je doorgaat, bijstuurt of stopt.

Wat je oplevert

Aan het eind van dit hoofdstuk heb je in ieder geval twee concrete documenten:

- **Een ingevuld Agent Design Canvas** Hierin staan het probleemstatement, de doelgroep en actoren, de value metric en ondersteunende KPI's, de scope met in- en out-of-scope, de invoer en uitvoer, relevante beperkin-

gen en waar mens in de lus noodzakelijk is. Ook leg je acceptatiecriteria en killcriteria vast, zodat besluitvorming niet op gevoel gebeurt maar op metingen.

- **Een korte bekostigingsschets** Je maakt een orde-grootteberekening van de kosten per taak en per maand, inclusief aannames over volumes, tokengebruik, externe API-tarieven en beheerinspanning. Je noteert bandbreedtes en de factoren die die bandbreedtes bepalen.

Met deze twee onderdelen kun je aan een beslisser in enkele minuten uitleggen wat je gaat doen, wat het mag kosten en wanneer je stopt.

Hoe je dit hoofdstuk gebruikt

Plan een korte werksessie met de belangrijkste betrokkenen: de eigenaar van het proces, iemand uit uitvoering of support, iemand die data kan aanleveren en iemand die over privacy en beveiliging adviseert. Reserveer negentig minuten. Neem een steekproef van recente cases of tickets mee, plus basisstatistieken zoals aantallen per week, doorlooptijden en foutpercentages.

Volg de hoofdstukstructuur in deze volgorde:

1. Lees de velden van het canvas door zodat je de betekenis per onderdeel kent.
2. Vul het canvas stap voor stap in. Begin bij het probleemstatement en de value metric. Beperk de scope expliciet en noteer afhankelijkheden.
3. Werk de invoer en uitvoer uit tot het niveau waarop je later kunt testen of de uitvoer structureel klopt.
4. Benoem beperkingen en bepaal waar menselijke controle nodig is, inclusief een fallback.
5. Leg KPI's, acceptatiecriteria en killcriteria vast. Zorg dat elk criterium meetbaar is binnen vier weken.
6. Maak de bekostigingsschets. Gebruik eenvoudige rekenpaden en vermeld aannames en bandbreedtes.

Rond af met een korte check: is het probleem concreet, is de waarde meetbaar, is de scope begrensd, zijn risico's voorzien van maatregelen en is duidelijk wie beslist? Als een van deze vragen met nee wordt beantwoord, herformuleer je het desbetreffende onderdeel voordat je doorgaat naar ontwerp of bouw.

Overzicht van het Agent Design Canvas

Deze paragraaf geeft een snel, gedeeld beeld van wat het canvas is en hoe je het document gebruikt in het verdere traject. Je ziet waar het canvas in de levenscyclus past en welke onderdelen je invult in de paragraaf hierna.

Doel en plaats in het boek

Het Agent Design Canvas is het startdocument voor elke agentcasus. Het voorkomt dat je te vroeg in tools of implementatie duikt en dwingt tot keuzes over waarde, scope en risico. In dit boek is het canvas het scharnier tussen idee en uitvoering: je gebruikt het resultaat uit hoofdstuk 1 om in hoofdstuk 2 de bouwstenen van je agent te kiezen, in hoofdstuk 3 de architectuur te bepalen en in hoofdstuk 5 en verder de prompts, evaluatie en operatie vorm te geven. Het canvas blijft daarna een levend document: bij wijzigingen in scope, KPI's of risico's werk je de laatste versie bij en leg je vast waarom.

Je vult het canvas in:

- bij start van een nieuw initiatief of wanneer een bestaand proces merkbaar knelt (wachttijden, fouten, hoge kosten);
- na een korte verkennende analyse, zodat je probleem en data kan onderbouwen met een steekproef;
- vóór je een proof-of-concept maakt, zodat je succescriteria vooraf vaststaan.

De betrokkenen bij het canvas zijn:

- de proceseigenaar met mandaat om scope, KPI's en budget te bepalen;
- een uitvoerende gebruiker die de dagelijkse praktijk kent;
- iemand met datatoegang die basisstatistieken kan leveren;
- een privacy- of securityadviseur die beperkingen kan duiden.

Onderdelen in één oogopslag

Het canvas bestaat uit twaalf velden die je consequent invult. Elk veld helpt om latere discussies te verkorten en beslissingen te objectiveren.

1. **Probleemstatement** – één zin met probleem, impact en processtap.
2. **Doelgroep en actoren** – primaire gebruikers, stakeholders, eigenaarschap en besluitvorming.
3. **Value metric en KPI's** – centrale waarde-indicator met ondersteunende meetpunten en meetfrequentie.

4 **ROI-hypothese** – baten, kosten en expliciete aannames met een eenvoudig rekenpad.
5 **Scope (in/out-of-scope)** – concrete afbakening van V1 met afhankelijkheden.
6 **Invoer en uitvoer** – bronnen, minimale datakwaliteit, uitvoerformaat en validatiecriteria.
7 **Beperkingen** – privacy, beveiliging, bewaartermijnen, data-locatie en auditvereisten met maatregelen.
8 **Mens in de lus** – momenten van menselijke controle met drempels, rol en doorlooptijd.
9 **Killcriteria** – objectieve stopcondities die een pauze of rollback triggeren.
10 **Acceptatiecriteria** – drempels voor "V1 is klaar" op kwaliteit, tijd en kosten.
11 **Evaluatieplan in het klein** – golden set, testmomenten, beoordelingswijze en rapportagevorm.
12 **Bekostigingsschets** – orde van grootte van kosten per taak en per maand, met bandbreedtes en aannames.

Gebruik de lijst als agenda voor je werksessie. Vul elk veld kort en concreet in. Noteer aannames, brondata en besluiten. Laat aan het einde iemand eigenaarschap nemen over het canvas, inclusief datum en versienummer. Zo blijft het document bruikbaar voor besluitvorming en audit.

Velden van het canvas

Deze paragraaf beschrijft elk veld van het Agent Design Canvas. Vul de velden kort en concreet in. Werk met bestaande data en spreek meetmomenten af. Elk veld eindigt idealiter in één beslispunt of drempelwaarde.

Probleemstatement

Formuleer in één zin welk probleem je oplost, waar in het proces het ontstaat en wat de impact is. Vermijd symptomen en benoem de processtap. Bijvoorbeeld: "Het supportteam besteedt te veel tijd aan handmatig routeren van e-mails, waardoor de first response time stijgt en tickets blijven liggen."

- Check: is de oorzaak procesmatig te beïnvloeden en binnen scope van een agent?

Doelgroep en actoren

Beschrijf wie de agent gebruikt of raakt. Noem primaire gebruikers, secundaire stakeholders en de eigenaar met mandaat. Leg vast wie beslist over livegang en wie het draaiboek beheert.

- Check: is duidelijk wie waarvoor verantwoordelijk is en wie goedkeurt?

Value metric en KPI's

Kies één value metric die het nut van de agent vangt. Voeg hooguit enkele ondersteunende KPI's toe. Leg bron en meetfrequentie vast. Voorbeelden voor value metric: first response time in minuten. Voorbeelden voor KPI's: juiste routering in percentage, doorstroom per uur, uitzonderingsratio in percentage.

- Check: is de value metric binnen vier weken aantoonbaar te verbeteren?

ROI-hypothese

Schets baten, kosten en aannames. Gebruik een eenvoudig rekenpad van metric naar uren of euro's. Noem bandbreedtes en de factoren die die bandbreedtes bepalen. Voor baten kijk je naar bijvoorbeeld bespaarde uren per week, vermeden fouten en mogelijke omzetimpact. Qua kosten komen tokens, externe API's, licenties, realisatie-uren en beheer in aanmerking. Denk bij aannames aan volumes, tarieven en verwacht succespercentage.

- Check: is het rekenpad herhaalbaar met nieuwe data?

Scope: in-scope en out-of-scope

Begrens V1 helder. Benoem afhankelijkheden. Noteer wat bewust niet gedaan wordt. In-scope zijn bijvoorbeeld e-mails met onderwerp 'Factuur'. Out-of-scope zijn meertalige tickets. Afhankelijkheid kan bijvoorbeeld CRM-tag 'debiteur' zijn.

- Check: kan een buitenstaander uitleggen waar V1 begint en eindigt?

Invoer en uitvoer

Specificeer bronnen, minimale datakwaliteit, gewenste uitvoer en validatie. Maak uitvoer bij voorkeur machineleesbaar met duidelijke velden. Specificeer voor invoer: kanaal, formaat, verplichte velden en datakwaliteitseisen. Voor uitvoer specificeer je formaat, bestemmeling, validatiecriteria en tolerantie.

- Check: kun je automatisch controleren of de uitvoer aan de structuur voldoet?

Beperkingen: naleving en risico's

Benoem privacy, beveiliging, bewaartermijnen, datalocatie en auditvereisten. Koppel aan elke beperking een maatregel en een eigenaar. Bijvoorbeeld: persoonlijk identificeerbare informatie (PII) beperken tot noodzakelijke velden. Verwerking binnen de EU. Bewaartermijn ruwe logs negentig dagen, auditmetadata één jaar.

- Check: is voor elke beperking duidelijk hoe je naleving aantoont?

Mens in de lus

Plaats bewuste interventies waar dat nodig is. Definieer trigger, drempel, rol, doorlooptijd en fallback. Beperk het aantal interventies. Bijvoorbeeld: bij vertrouwen lager dan 0,8 vraagt de agent om goedkeuring in Teams. De supportlead beslist binnen vijftien minuten.

- Check: is de interventielast proportioneel en meetbaar?

Killcriteria

Leg objectieve stopcondities vast die een pauze of rollback vereisen. Formuleer drempels en periode. Voorbeelden: structuurnaleving lager dan vijfennegentig procent gedurende drie dagen, kosten per ticket hoger dan € 1,50 gedurende één week of onjuiste routering hoger dan tien procent op een steekproef van honderd.

- Check: zijn killcriteria automatisch te monitoren met een alert?

Acceptatiecriteria

Bepaal wanneer V1 'klaar' is. Werk met meetbare drempels op kwaliteit, tijd en kosten. Neem documentatie op als eis. Bijvoorbeeld: juiste routering ten minste negentig procent op validatieset; first response time dertig procent lager dan baseline; maximaal één procent schemafouten in uitvoer; draaiboek en evaluatieverslag aanwezig.

- Check: zijn de drempels haalbaar binnen de scope van V1?

Evaluatieplan in het klein

Plan dataset, testmomenten, beoordelingswijze en rapportage. Houd het licht, maar reproduceerbaar. Minimaal definieer je een *golden set* van vijftig representatieve gevallen met juiste uitkomst. Meetmomenten zijn pre-launch, dag 1,

week 1, week 4. Ter beoordeling een tweede reviewer bij twijfelgevallen. En als rapportage een tabel met KPI's en korte notities.

- Check: kun je dezelfde set later gebruiken voor regressietests?

Bekostigingsschets

Geef orde van grootte van kosten per taak en per maand, met bandbreedtes en aannames. Noteer tarieven en volumes expliciet. Bereken:

- tokens per taak maal prijs per token;
- externe API's per taak maal tarief;
- aantal taken per maand;
- beheeruren per maand maal tarief.

Bijvoorbeeld: € 0,03-0,06 per taak bij 50.000 taken per maand geeft circa € 1500-3000 per maand, exclusief acht beheeruren.

- Check: is helder welke aannames de bandbreedte bepalen en wie ze bijwerkt?

Van idee naar canvas in zes stappen

Je vertaalt een ruw idee in een beslisbaar plan en een ingevuld canvas, met heldere checkpoints en valkuilen. De benodigdheden daarvoor zijn de proceseigenaar (met mandaat), uitvoerend gebruiker, datacontact, privacy/security-adviseur, steekproef van recente cases en basisstatistieken (aantallen, doorlooptijden, foutpercentages).

Stap 0 – Voorbereiden (15 minuten)

1. Verzamel een steekproef van vijftig recente gevallen en noteer basisstatistieken.
2. Leg aanwezigen en rollen vast (wie beslist, wie beheert het draaiboek?).
3. Bepaal de duur van deze sessie (negentig minuten) en het doel (canvas plus bekostigingsschets).

- Controleer: is de steekproef representatief en zijn rollen helder vastgelegd?
- Valkuil: starten zonder data of zonder beslisser aan tafel.

Stap 1 – Probleem en waarde scherpstellen (15 minuten)

1. Formuleer één zin voor het probleemstatement (processtap plus impact).

2 Kies één value metric en maximaal drie ondersteunende KPI's.
3 Leg bron en meetfrequentie vast.

- Controleer: kun je binnen vier weken een verbetering in de value metric waarnemen?
- Valkuil: een symptoom kiezen ("te weinig mensen") in plaats van een procesoorzaak.

Stap 2 – Scope begrenzen (10 minuten)

1 Noteer in-scope en out-of-scope concreet.
2 Leg afhankelijkheden en noodzakelijke integraties vast.
3 Bepaal expliciet wat in V2 thuishoort.

- Controleer: kan een buitenstaander de grenzen van V1 uitleggen?
- Valkuil: scope-kruip door stille uitbreidingen tijdens de sessie.

Stap 3 – Invoer en uitvoer specificeren (15 minuten)

1 Beschrijf per bron: kanaal, formaat en verplichte velden.
2 Definieer de uitvoer: formaat (bijvoorbeeld JSON), bestemmeling en validatiecriteria.
3 Leg tolerantiegrenzen vast (bijvoorbeeld maximaal één procent schemafouten).

- Controleer: kun je de uitvoer automatisch valideren op structuur?
- Valkuil: vrije tekst als eindproduct waar gestructureerde data nodig is.

Stap 4 – Risico's en mens in de lus (15 minuten)

1 Benoem privacy- en beveiligingsrisico's, inclusief datalocatie en bewaartermijnen.
2 Plaats een mens in de lus waar nodig: trigger, drempel, rol, doorlooptijd en fallback.
3 Koppel per risico een maatregel en een eigenaar.

- Controleer: is naleving aantoonbaar en de interventielast proportioneel?
- Valkuil: algemene statements ("we letten op privacy") zonder operationele invulling.

Stap 5 – Acceptatie- en killcriteria (10 minuten)

1. Definieer drempels voor "V1 is klaar" (kwaliteit, tijd, kosten).
2. Leg killcriteria met periode vast (bijvoorbeeld drie dagen, één week).
3. Leg meet- en alertingmethode vast.

- Controleer: zijn alle criteria meetbaar en te monitoren met een alert?
- Valkuil: vage grenzen ("te duur", "te foutgevoelig").

Stap 6 – Bekostigingsschets (10 minuten)

1. Bereken kosten per taak: tokens + externe API's + runtime.
2. Schaal naar maand: taken per maand × kosten per taak.
3. Voeg beheeruren en licenties toe; noteer aannames en bandbreedtes.

- Controleer: is duidelijk welke aannames de range bepalen en wie ze bijwerkt?
- Valkuil: schijnprecisie: twee cijfers achter de komma zonder onzekerheidsmarge.

Afronden en vastleggen (10 minuten)

1. Controleer alle velden op consistentie en haalbaarheid.
2. Wijs een eigenaar aan voor het canvas en voor de KPI-meting.
3. Versienummer en datum toevoegen; beslis noteren: go, bijsturen of pauzeren.

Het resultaat van deze oefening is een ingevuld canvas met versienummer, een korte bekostigingsschets en expliciete afspraken over meting, alerting en eigenaarschap.

Voorbeeld ingevuld canvas

Deze voorbeeldcasus laat zien hoe een compact, beslisbaar canvas eruitziet. Gebruik het als referentie bij je eigen invulling.

Samenvatting casus

Middelgroot softwarebedrijf met een supportteam dat dagelijks circa tweehonderd e-mails ontvangt. Wachttijden lopen op, klanten melden onduidelijke doorlooptijden. Het doel is het automatisch routeren en verrijken van inkomende supportmails zodat de first response time daalt en tickets sneller bij de juiste eigenaar komen.

Canvas

Velden	Kerninhoud
Probleemstatement	Inkomende e-mails worden handmatig gelezen en toegewezen, wat leidt tot vertraging en verkeerd gerouteerde tickets.
Doelgroep en actoren	Primaire gebruikers: supportmedewerkers. Stakeholders: operations, sales. Eigenaar: supportmanager. Beslisser: head of operations.
Value metric en KPI's	Value metric: first response time in minuten. KPI's: juiste routering in percentage, doorstroom per uur, uitzonderingsratio in percentage.
ROI-hypothese	Baten: vijftien uur per week minder handmatig werk, minder misroutes. Kosten: modellen en API's circa € 1500-3000 per maand bij 50.000 taken, exclusief beheer. Aannames: zeventig procent van de e-mails is automatisch te routeren, gemiddelde ticketlengte 1200 tokens.
Scope (V1)	In-scope: Nederlandstalige factuur- en accountvragen via e-mail. Out-of-scope: meertalige tickets, escalaties, refunds. Afhankelijkheden: CRM-tag 'debiteur', mailbox 'support@example.com'.
Invoer en uitvoer	Invoer: onderwerp, afzender, bodytekst, ticketnummer indien aanwezig. Uitvoer: JSON met category, priority, assignee, confidence, plus samenvatting (maximaal 200 tekens).
Beperkingen	PII minimaliseren, verwerking binnen EU, bewaartermijn ruwe logs negentig dagen, auditmetadata één jaar.
Mens in de lus	Bij vertrouwen lager dan 0,8 vraagt de agent goedkeuring in Teams. Supportlead beslist binnen vijftien minuten.

Velden	Kerninhoud
Killcriteria	Structuurnaleving lager dan vijfennegentig procent gedurende drie dagen. Kosten per ticket hoger dan € 1,50 gedurende één week. Onjuiste routering hoger dan tien procent op een steekproef van honderd.
Acceptatiecriteria (V1 klaar)	Minimaal negentig procent juiste routering op validatieset. First response time dertig procent lager dan baseline. Maximaal één procent schemafouten in uitvoer. Draaiboek en evaluatieverslag aanwezig.
Evaluatieplan (klein)	Golden set: vijftig gelabelde e-mails. Meetmomenten: pre-launch, dag 1, week 1, week 4. Dubbelcheck twijfelgevallen.
Bekostigingsschets	€ 0,03-0,06 per taak bij 50.000 taken per maand ca. € 1500-3000 per maand, exclusief acht beheeruren.

Belangrijkste keuzes en rationale

- Machineleesbare uitvoer: JSON maakt automatische validatie mogelijk.
- Mens-in-de-lusdrempel: 0,8 balanceert kwaliteit en doorlooptijd.
- Smalle scope: eerst factuur- en accountvragen om datakwaliteit en patronen te stabiliseren.
- Killcriteria vooraf: voorkomt dat je doorsleutelt bij structurele fouten.

Overzichtstabel met drempels

- Juiste routering: = 90 procent (validatieset).
- Structuurnaleving: = 99 procent (schemafouten = 1 procent).
- Kosten per ticket: = € 1,50 (rollend weekgemiddelde).
- Interventieratio: = 20 procent in week vier.

Checklists en valkuilen

Voordat je naar ontwerp of bouw gaat, toets je het canvas op een paar wezenlijke punten. Begin bij de kern: staat het probleem scherp, in één zin, gekoppeld aan een herkenbare processtap en een aantoonbare impact? Als die zin nog

voelt als een symptoombeschrijving (bijvoorbeeld “we hebben te weinig mensen”), herschrijf dan tot je de procesoorzaak te pakken hebt. Kijk vervolgens naar de value metric. Deze moet niet alleen logisch zijn, maar ook meetbaar met bestaande data en zichtbaar kunnen verbeteren binnen vier weken. Leg daarom bron en meetfrequentie vast en wijs een rol aan die de meting uitvoert.

Daarna begrens je de scope. Noteer concreet wat V1 wel en niet doet en zet afhankelijkheden erbij. Een smalle, heldere scope maakt het hoofdstuk beslisbaar en voorkomt scope-kruip. Controleer of de uitvoer structureel te valideren is. Waar het kan, kies je voor machineleesbare uitvoer met een schema en een kleine tolerantie voor fouten; vrije tekst is zelden een goed eindproduct als er vervolgacties van systemen nodig zijn.

Risico’s verdienen dezelfde scherpte. Beschrijf voor privacy, beveiliging, datalocatie en bewaartermijnen niet alleen het risico, maar ook de maatregel en de eigenaar. Zo kun je naleving aantonen in plaats van het te veronderstellen. Plaats mens in de lus waar dat nodig is en niet vaker. Bepaal een drempel, leg de rol vast die beslist, spreek een doorlooptijd af en zorg voor een fallback. Een schaars en goed gekozen interventiemoment bewaakt de kwaliteit zonder de doorstroming te blokkeren.

Rond het canvas af met drempels die richting geven aan besluitvorming. Formuleer acceptatiecriteria voor kwaliteit, tijd en kosten, en leg killcriteria vast met een duidelijke periode waarover je ze beoordeelt. Organiseer monitoring en alerts zodat je niet op gevoel hoeft te handelen als het spannend wordt. Neem ten slotte een beknopte bekostigingsschets op. Reken van taak naar maand en vermeld openlijk de aannames en bandbreedtes waarop de uitkomst rust.

Let op de valkuilen die we vaak zien. Teams schieten nog wel eens een symptoom aan, kiezen een metric zonder beschikbare data of nemen te veel tegelijk op in V1. Ook PII-afdekking en bewaartermijnen blijven soms impliciet, met alle risico’s van dien. Een andere klassieker is het vergeten van een rollbackpad: zonder killcriteria blijf je itereren, ook als het niet werkt. Tot slot ondermijnt schijnprecisie de besluitvorming: laat bandbreedtes zien, wees expliciet over aannames en wijs een eigenaar aan voor monitoring en naleving.

Mini-evaluatie en besluit

Na het invullen van het canvas wil je snel beslissen: doorgaan, bijsturen of pauzeren. Deze mini-evaluatie is bewust lichtgewicht. Je gebruikt de gegevens die je zojuist hebt vastgelegd, zonder nieuwe analyses. Het doel is helderheid in één korte bespreking: voldoet V1 aan de afgesproken drempels en voorwaarden, of niet?

Go/no-go op V1

Begin met de kern: staat de waardeverwachting nog overeind tegen de kosten en risico's die je hebt benoemd? Loop de drempels langs zoals je ze in de paragraaf *Velden van het canvas* hebt vastgelegd. Kijk eerst naar de value metric en de drie belangrijkste KPI's. Zijn de beoogde verbeteringen realistisch binnen vier weken, en is de meetbaarheid geborgd (bron, frequentie, rol)? Controleer daarna de structurele voorwaarden: is de scope van V1 haalbaar, zijn invoer en uitvoer voldoende gespecificeerd, en is de uitvoer controleerbaar op structuur? Tot slot weeg je de beperkingen en de mens in de lus: zijn privacy- en beveiligingsmaatregelen operationeel, en is de interventielast acceptabel?

Een praktische manier om het gesprek te sturen is het besliskwadrant: links de waarde (hoog/laag), onder de risico's en kosten (laag/hoog). Plaats de casus en bespreek waarom. Casussen met hoge waarde en lage risico's/kosten gaan door. Casussen met hoge waarde maar hoge risico's of onduidelijke kosten vragen om een gerichte bijsturing (bijvoorbeeld: drempel aanscherpen, scope verkleinen, extra mitigerende maatregel). Casussen met lage waarde en hoge risico's pauzeer je, tenzij er een overtuigend pad naar een betere positie ligt.

Sluit af met een expliciet besluit en een korte notitie van de rationale. Noteer ook de killcriteria nogmaals en check of monitoring en alerts zijn ingericht. Zo voorkom je grijze gebieden na de start.

Volgende stap

Bij een go neem je drie documenten mee naar de vervolghoofdstukken:

1. Het ingevulde canvas (met versienummer, datum en eigenaar). Dit document is het vertrekpunt voor de ontwerpkeuzes in hoofdstuk 2 (bouwstenen) en hoofdstuk 3 (architectuur).
2. De uitvoerspecificatie (velddefinities en validatieregels). Deze tekst gebruik je direct in hoofdstuk 5 bij het opzetten van prompts met gestructureerde uitvoer en validatie.

3 De evaluatie- en kostenafspraken (KPI's, meetmomenten, alerting, bandbreedtes en aannames). Je hergebruikt ze in hoofdstuk 8 (evaluatie en testen) en hoofdstuk 11 (kosten en performance) om voortgang en efficiëntie te bewaken.

Bij een bijsturen formuleer je één gerichte aanpassing die de positie in het besliskwadrant verbetert (bijvoorbeeld scope versmallen of een extra guardrail). Beperk bijsturing tot één iteratie en plan direct een nieuwe mini-evaluatie.

Bij een pauze archiveer je het huidige canvas met de reden van het besluit. Leg vast welke data of voorwaarden ontbreken en wanneer herbeoordeling zinvol is (bijvoorbeeld na nieuwe brondatasets of na het beschikbaar komen van een integratie). Zo blijft het werk herbruikbaar en transparant, ook als je nu niet verder gaat.

Resulterend materiaal

Aan het einde van dit hoofdstuk ligt er geen stapel losse notities, maar een compacte set documenten waarmee je verder kunt. Het eerste en belangrijkste document is het ingevulde Agent Design Canvas. Zorg dat de versie, de datum en de eigenaar erop staan. Het canvas is vanaf nu het referentiepunt voor alle keuzes: waarom we dit doen, voor wie, met welke waarde, binnen welke grenzen en onder welke voorwaarden. Bij wijzigingen werk je dezelfde versie bij en noteer je kort waarom; zo blijft de rode draad zichtbaar voor iedereen die aanhaakt.

Het tweede document is de korte bekostigingsschets. Hierin vat je de aannames samen, reken je de kosten per taak en per maand door en geef je de bandbreedte aan. Het document moet in één oogopslag laten zien waar de bedragen vandaan komen: tokens, externe API's, licenties en beheeruren. Benoem expliciet de factoren die de bandbreedte bepalen, zoals volumes, gemiddelde invoerlengte of succespercentage. Zodra deze aannames veranderen, pas je de schets aan en koppel je de impact terug aan de eigenaar van de KPI's.

Het derde document is de log van beslissingen. Dit zijn geen uitgebreide notulen, maar een korte tijdlijn van de belangrijkste keuzes en de rationale erachter. Denk aan de gekozen value metric, de drempel voor mens in de lus, de afgesproken killcriteria en de grenzen van V1. Door deze log te onderhouden, kun je later verantwoorden waarom een pad is gekozen en kun je sneller terug-

draaien als omstandigheden veranderen. De log ondersteunt bovendien audit en kennisoverdracht wanneer het team wijzigt of groeit.

Met deze drie documenten ben je klaar voor het vervolg: het canvas stuurt de selectie van bouwstenen en architectuur, de uitvoerspecificatie en evaluatieafspraken voeden het prompt- en testwerk, en de bekostigingsschets houdt verwachtingen over middelen en impact realistisch. Je kunt nu gecontroleerd de stap naar hoofdstuk 2 en verder zetten.

Hands-on opdracht

Deze opdracht laat je het canvas direct toepassen op een eigen proces. Kies een proces dat tastbaar is en waar je binnen vier weken een meetbare verbetering kunt realiseren. Werk bij voorkeur met recente, echte voorbeelden zodat je aannames kunt toetsen.

Opdrachtbeschrijving

Selecteer één proces waarin herhaling en wachttijd zichtbaar zijn (bijvoorbeeld intake van e-mails, verrijking van leads of interne verzoeken). Vul het Agent Design Canvas volledig in. Begin met het probleemstatement, kies één value metric en voeg maximaal drie KPI's toe. Beleg eigenaarschap, begrens de scope van V1, specificeer invoer en uitvoer, benoem beperkingen met maatregelen en leg mens in de lus uit waar dat nodig is. Formuleer acceptatie- en killcriteria en maak een korte bekostigingsschets op basis van een kleine steekproef.

Verwachte uitkomst

Aan het einde heb je een consistent, beslisbaar canvas en een beknopte kostenraming. Het canvas bevat een probleem in één zin, meetbare waarde, een heldere scope en een uitvoerspecificatie die later automatisch te valideren is. De kostenraming maakt aannames expliciet en toont een bandbreedte per taak en per maand.

Beoordelingscriteria

Beoordeel je resultaat op vier punten: volledigheid (alle velden ingevuld en logisch), meetbaarheid (metrics, meetmomenten en bron vastgelegd), haalbaarheid (scope V1 is uitvoerbaar binnen de gestelde tijd) en risicoafdekking (privacy, beveiliging en mens in de lus zijn operationeel ingericht, met killcriteria en rollback). Leg eventuele open punten vast en plan gericht welke je vóór hoofdstuk 2 nog dichtzet.