

4.3 Context layers Context wordt opgebouwd in lagen. Elke laag bestaat uit concrete slots/velden die de engine kan verzamelen en vervolgens gebruiken in coaching en roleplay. Layer Doel Voorbeeldslots base Personalisatie en setting sector, product, klant_type, verkoopkanaal, ervaring scenario Gespreksrealiteit (start van gesprek) afspraak_type, rol gesprekspartner, lastige_openingsvragen, DMU value_map Probe/Impact laten landen op baten pains, baten, koopredenen_recent, usp_naar_baat_map objection_bank Realistische afritten/closing bezwaren_typisch, twijfels_typisch, uitstelredenen_typisch offer_map Fase 3/4 coherent maken oplossing_kern, voordelen, bewijsvoering, prijsrange ---> Deze lijst gaan we nog verder moeten aanvullen, toch? Welke vragen in fase 1, 2, 3 en 4? bijvoorbeeld? en hoe reageren. en er zijn meer houdingen van de klant dan dit. Het moet ook evenwichtig zijn: het evenwicht tussen context gathering en coaching en rollenspel en feedback. Ik heb de csv filled bekeken maar hij is niet filled. enkel de vragen zijn filled en als ik al die vragen bekijk dan krijg ik het als verkoper al benauwd. zooveel informatie. dat moeten we niet weten om goed te kunnen coachen of een geloofwaardige klant te spelen. we zijn het hier al deftig op het spoor. maar we moeten ook weten waar we echt goed in zijn in vergelijking met de concurrentie en waar mensen op afhaken (terecht en onterecht). sterktes van ons product en zwaktes. swot eigenlijk. of mis ik iets in het plan?

Thought for 9m 48s

oh en wat met de volgorde: nu staat de lijst zo opgemaakt eerste de fase en dan de deeltechnieken. dit zou er toe leiden dat ik eerst een rollenspel speel over fase 2 en dan de vragen en coaching sessie krijg over 2.1 en over 2.1.1. terwijl rollenspel duidelijk op het einde moet komen. zit dat er inervat? wanneer?

Got it – ik ben er op ingegaan en heb je aanvullingen en context meegenomen in de analyse.

Je hebt drie volledig terechte punten:

1. **Ja, die context-layers (en vooral de “slotlijst”) moeten inhoudelijk rijker kunnen** (SWOT/competitors/afhaakredenen), maar...
2. **Nee, we mogen dat niet vertalen naar “nog 15 extra contextvragen upfront”** — anders creëer je exact het benauwende gevoel dat jij beschrijft.
3. **En je volgorde-opmerking is cruciaal:** als je curriculum/overzicht “fase → dan deeltechnieken” is, dan mag dat niet impliceren dat het *capstone*-rollenspel (bv. “fase 2 full EPIC”) vóór de microbouwstenen komt.

Hieronder geef ik een concrete aanvulling op het plan: (A) “lean context layers” + (B) voorbeeldvragen per fase + (C) volledige houdingen + (D) balansregels + (E) oplossing voor de volgorde (in config).

A) Context layers: uitbreiden, maar in “thin-slice” + progressive disclosure

Je huidige lagen zijn goed als **conceptueel model**, maar je mist inderdaad nog minstens één dimensie: **positionering** (SWOT, concurrentie, afhaakredenen, disqualifiers). Alleen: die hoort niet per se “required” te zijn vanaf minuut 1.

1) Maak per layer een “Minimum set” en een “Nice-to-have set”

In plaats van “layer = 10 vragen”, maak je “layer = slots”, en per slot een prioriteit. Dan kan de orchestrator met een **context budget** werken.

Hier is een werkbaar voorstel (lean):

base (minimum voor coaching)

Minimum slots (2–5 bullets, geen interview):

- sector
- product/dienst (1 zin)
- klant_type (ICP in 1 zin)
- verkoopkanaal
- ervaring (optioneel)

Nice-to-have:

- dealgrootte-range / prijsrange
 - regio/taal
 - maturity van aanbod (nieuw/volwassen)
-

scenario (minimum voor realistische start)

Minimum slots:

- afspraak_type (cold/warm/referral/inbound)
- rol gesprekspartner (DM/influencer/user)
- context van “waarom zitten we samen”
- 1 typische lastige openingsvraag

Nice-to-have:

- DMU: wie beslist nog mee
 - timing/urgentie indicatie
 - huidige oplossing/alternatief
-

value_map (minimum om Probe/Impact “te laten landen”)

Minimum slots (thin):

- top pain (1)
- gewenste uitkomst/baten (1)
- wat doen ze nu (huidige workaround)

Nice-to-have (rich):

- top 3 pains
- top 3 baten
- koopredenen_recent (top 3)
- beslissingscriteria (top 3)
- usp_naar_baat_map (3 koppelingen)

Belangrijk: **value_map hoeft niet vooraf volledig ingevuld**. Een groot deel kan je **uit het Explore/Probe rollenspel extraheren** en als artifact opslaan.

objection_bank (minimum om “afritten” realistisch te maken)

Minimum slots (thin):

- 1 typisch bezwaar
- 1 typische twijfel
- 1 typische uitstelreden

Nice-to-have:

- top 5 bezwaren
- top 5 twijfels
- top 5 uitstelredenen
- 1 “angst/risico”-patroon (wat zijn ze bang om te verliezen?)

Ook hier: veel bezwaren komen vanzelf naar boven **in fase 3/4 rollenspel**, dus je hoeft dit niet volledig vooraf te bevragen.

offer_map (minimum om fase 3/4 coherent te maken)

Minimum slots:

- oplossing_kern (1 zin)
- 3 belangrijkste voordelen (bullets)
- 1 bewijsstuk (case, referentie, metric, demo)
- prijsrange/contractvorm (globaal)

Nice-to-have:

- implementatie-aanpak (in 3 stappen)
- differentiatoren (top 3)

- voorwaarden/constraints (wat kan je niet)
 - “next step menu” (demo, workshop, voorstel, pilot)
-

2) Nieuwe laag die jij terecht aanbrengt: Positioning/SWOT

Dit is wat jij bedoelt met “waar zijn we écht goed vs concurrentie” + “waar haken mensen af (terecht/onterecht)”.

Ik zou dit niet “SWOT” noemen in de chat (te consultant-y), maar wel als **positioning_map** of **fit_map** modelleren.

positioning_map (sterktes/zwaktes/concurrentie/afhaakredenen)

Minimum slots:

- top 3 sterktes (waarom winnen we)
- top 2 zwaktes/trade-offs (waarom verliezen we terecht)
- top 3 “afhaakredenen-onterecht” (mispercepties)
- top 3 concurrenten / alternatieven (desnoods categorieën)

Waarom dit belangrijk is

- Voor geloofwaardige klanten: “Wij twijfelen omdat concurrent X...”
- Voor coaching: wanneer moet je **disqualificeren** vs wanneer is het **misperceptie** die je kan rechtekken.

Let op: dit moet **optioneel** zijn in vroege fases, anders maak je onboarding zwaar.

B) Welke vragen per fase: concreet, maar met “context budget”

Je punt “zoveel vragen = benauwd” is 100% juist. De oplossing is: **max 2–3 contextvragen**, dan pas coaching/waarde. En alleen verdiepen wanneer de gebruiker expliciet “dieper rollenspel” wil.

Hier zijn voorbeeldvragen per fase (minimum) die al 80% dekken:

Fase 1: Opening (koopklimaat + GA + instap)

Doel: realistische start van het gesprek, zonder overload.

Context (max 3 vragen):

1. “Wat verkoop je (1 zin) en aan wie verkoop je het meestal?”
2. “Wat voor afspraak is dit meestal: inbound, cold, referral... en via welk kanaal?”

3. “Welke lastige openingsvraag krijg je in de eerste 2 minuten het vaakst?”

Daarna coaching (geen extra context-interview):

- Samen een Gentleman's Agreement formuleren in hun woorden.
- Instapvraag herwerken naar hun sector/product.
- Mini-drill: 2 varianten hardop formuleren.

Klantreacties (houdingen) in fase 1:

- vooral: **vaag / ontwijkend / vraag / positief / negatief**
 - nog niet: twijfel/bezwaar/uitstel/angst (die horen pas na een aanbod/afsluitmoment).
-

Fase 2: Ontdekkingsfase (EPIC)

Doel: Explore/Probe/Impact laten werken met “voldoende” value-context, maar niet perfect.

Context (max 3 vragen) – thin slice:

1. “Noem 1 typische aanleiding/pijn waarom klanten bij jou uitkomen.”
2. “Wat willen ze op het einde écht bereikt hebben (de baat)?”
3. “Wat is 1 bezwaar dat al vroeg in het gesprek opduikt?”

Daarna:

- Coach op vraagvormen en luisterhouding (micro), of meteen Explore-roleplay (afhankelijk van je unlock-regels, zie verder).
- Tijdens/na roleplay laat je de engine **discovery_brief** genereren (artifact), i.p.v. vooraf 15 vragen te stellen.

Klantreacties in fase 2:

- nog steeds vooral: **vaag / ontwijkend / vraag / negatief / positief**
 - het “echte bezwaar” als afsluit-houding hoort later; maar “vroeg bezwaar” kan als *interruption* voorkomen en dan parkeer je het (terzijde schuiven).
-

Fase 3: Aanbevelingsfase

Doel: aanbod logisch laten volgen uit discovery.

Gate-idee: fase 3 geïntegreerd rollenspel is pas geloofwaardig met een **discovery_brief**. Maar die hoeft niet van tevoren “ingevuld”, die kan:

- uit fase 2 roleplay komen, of
- via een “mini recap” in 90 seconden.

Mini recap (max 3 vragen als discovery_brief ontbreekt):

1. “Wat was de kernpijn?”
2. “Wat is de gewenste uitkomst + belangrijkste criterium?”
3. “Wat is hun grootste twijfel/risico?”

Daarna:

- offer_map “thin”: 1 kernoplossing + 3 voordelen + 1 bewijs.
- Roleplay start vaak best op **commit** (je doet alsof discovery gebeurd is).

Klantreacties in fase 3:

- hier mogen **twijfel/bezwaar/uitstel/angst** echt opduiken (zeker na aanbod).
-

Fase 4: Afsluiten + bezwaren

Doel: omgaan met twijfel/bezwaar/uitstel/angst + next step.

Context (max 2 vragen):

1. “Wat is typisch de gewenste next step (demo, voorstel, pilot, workshop)?”
2. “Wat is de meest voorkomende ‘afhaakreden’ bij het closing moment?”

Daarna:

- drills per houding: twijfel, bezwaar, uitstel, angst.
 - feedback na roleplay.
-

C) “Er zijn meer houdingen dan dit” — klopt, en ze zijn al gedefinieerd

In jouw project zitten **9 klanthoudingen** (niet 3):

positief, negatief, vaag, ontwijkend, vraag, twijfel, bezwaar, uitstel, angst.

Belangrijk voor orkestratie:

- In fase 1–2: vooral de eerste vijf.
- In fase 3–4: ook de laatste vier (ze vereisen een “trigger”: aanbod/afsluiting).

Dit is exact wat je evenwicht bewaakt: je simuleert niet te vroeg closing-gedrag wanneer je nog in Explore zit.

D) Evenwicht tussen context gathering, coaching, roleplay, feedback

Als je dit niet expliciet maakt, wordt context gathering al snel een “intake interview”.

Ik stel 3 harde regels voor (die je als `flow_rules` kan opnemen):

1. **Max 2 contextvragen na elkaar**
Daarna altijd: samenvatten + waarde leveren (coach tip / voorbeeldzin / mini-drill).
2. **Context budget per sessie**
Bijvoorbeeld: **max 5 contextvragen totaal**, tenzij gebruiker expliciet “dieper” vraagt.
3. **Deeper layers pas op het moment dat ze renderen**
 - `positioning_map` pas echt nodig in fase 3/4 (of wanneer concurrentie ter sprake komt)
 - `objection_bank` groeit best organisch uit roleplays

Dit voorkomt dat de verkoper “benauwd” wordt, en maakt de chat menselijk.

E) Je volgordeprobleem: je hebt gelijk — en dit zat niet expliciet afgedwongen in V3

Je vraag:

“Nu staat de lijst fase → deeltechnieken. Dat zou leiden tot eerst rollenspel fase 2 en pas daarna coaching op 2.1/2.1.1. Terwijl rollenspel op het einde moet komen. Zit dat erin? Wanneer?”

Antwoord

- In de originele `coach_overlay_v3.json` zat dit niet hard afgedwongen.
- Daardoor kan een UI/flow die “van boven naar beneden” gaat inderdaad verkeerd uitkomen.

Oplossing (configmatig): “roleplay_unlock” op bundles en capstones

Ik heb daarom een verbeterde versie gemaakt die expliciet kan afdwingen:

- **1 (fase 1 capstone):** roleplay pas aanbieden nadat **1.1–1.4** minstens 1x geoefend zijn
- **2.1 (Explore bundle roleplay):** pas aanbieden nadat **2.1.1–2.1.6** minstens 1x geoefend zijn
- **2 (fase 2 full EPIC capstone):** pas aanbieden nadat **2.1–2.4** minstens 1x geoefend zijn
- idem voor fase 3 en 4

En ik heb ook een `sequence_policy` toegevoegd (ranking micro → drill → integrated) zodat je UI/next-step-logica consistent kan zijn.

Je kan die meteen downloaden:

- Download `coach_overlay_v3_1.json`

Wanneer komt het rollenspel dan?

- Op twee niveaus:
 1. **Binnen één techniek-sessie:** altijd op het einde (COACH_CHAT_THEN_ROLEPLAY)
 2. **Binnen het curriculum:** pas “unlocked” nadat de microbouwstenen geoefend zijn (roleplay_unlock)

Dat is exact wat jij bedoelt met “rollenspel duidelijk op het einde”.

Wat mis je nog in het plan?

Niet inhoudelijk, maar structureel zijn er nog 2 dingen die je best toevoegt:

1. **positioning_map / fit_map als optionele layer** (jouw SWOT-punt)
2. **unlock/sequence policy** (om het “fase eerst” overzicht niet te laten bepalen dat je te vroeg roleplay doet)

Als je wilt, kan ik nu ook heel concreet voorstellen:

- welke technieken/bundles een unlock nodig hebben (niet alleen 2.1 en 2),
- welke layers *minimum* vs *optional* moeten zijn per practice_unit,
- en hoe je dat in de orchestrator runtime omzet naar “toon volgende stap” zonder dat je 20 vragen stelt.

DEVELOPER MODE

ChatGPT can make mistakes. OpenAI doesn't use Stéphane Van de Velde's Workspace workspace data to train its models.