GreenFox-bot — advies op de 4 coaching-vragen (Fable 5)

> Dit is dezelfde opdracht als het eerdere advies, maar dan onafhankelijk gegenereerd door Fable 5 op basis van exact dezelfde twee bronnen (Plaud-transcript + Renzo's mail), zodat Jamal het kan vergelijken met de Sonnet-5-versie.

Coaching-advies GreenFox-bot — reactie op de vier vragen uit Renzo's mail

---

Wat ik uit het gesprek van 4 juli meeneem

Drie dingen uit jullie eigen gesprek stuur ik bewust mee in dit advies:

- **Het kernproces hangt aan één persoon.** Dat is jullie grootste risico. De bot moet dus niet alleen helpen, maar ook kennis vastleggen die nu in hoofden zit.
- **Weerstand is reëel.** Mensen voelen zich snel gecontroleerd. Junior intercedenten met zelfoverschatting nemen feedback slecht aan. De bot moet dus voelen als een hulpje van de medewerker zelf, nooit als een controle-instrument van de leiding.
- **Klantcommunicatie wil je uiteindelijk deels automatiseren, met checks.** Jullie guardrail "alles blijft concept" is daar precies het goede antwoord op. Houd die keihard vast.

---

a. Technische opzet van de GreenFox-bot op eigen server

Kernprincipe: één motor, veel profielen. Bouw niet per medewerker een aparte bot. Bouw één centrale bot-runtime, en maak per medewerker alleen een eigen map met configuratie.

Concreet:

- **Eén agent-runtime op de server** (bijvoorbeeld een agent-framework rond een LLM-API, of een lokaal model als jullie dat aankunnen). Alle persoonlijke bots zijn instanties van

dezelfde motor.

- **Per medewerker één werkmap** met vier bestanden: `profiel.md` (wie is dit, rol, stijl), `taken.md` (wat doet de bot voor deze persoon), `geheugen.md` (wat de bot leert over voorkeuren) en een map `concepten/` (alle output).
- **Eén gedeelde kennisbasis** (het Proceshuis, zie vraag b) die alle bots read-only lezen. Zo fix je een fout op één plek voor iedereen.
- **Alles onder versiebeheer (git) op de server.** Elke wijziging aan prompts, profielen en kennis is terug te draaien en te herleiden. Dit is ook jullie antwoord op guardrail 4 (zelfstandigheid): alle geleerde lessen staan als tekst in jullie eigen repo, niet in mijn hoofd.
- **Interface: chat, geen aparte app.** Jullie zitten in de Microsoft-omgeving met Intune en Fairphones. Een simpele webchat op het interne netwerk (of Teams-koppeling later) werkt op laptop én telefoon zonder app-beheer-gedoe.
- **Guardrails technisch afdwingen, niet alleen beloven:**
- De bot heeft géén verzendrechten. Geen mail-API, geen koppeling naar buiten. Output landt altijd in `concepten/`, een mens verstuurt.
- De bot heeft géén toegang tot Carerix/Easyflex-data in fase 1. Overzichten (contracten, subsidies) worden als export aangeleverd door de medewerker of Hamza. Koppelingen komen pas later, bewust.
- Alle bot-acties worden gelogd. Alleen Hamza heeft admin-rechten op de server. Anderen dienen wensen in via een whiteboard-lijst (jullie guardrail 7).
- **EU-compliance:** kies een model-aanbieder met EU-dataverwerking of draai lokaal. Omdat kandidaat- en klantdata toch niet naar het model gaan in fase 1, is het risico klein, maar leg de keuze nu al vast in een one-pager, dan hoef je die discussie nooit opnieuw te voeren.

---

b. Het Proceshuis als betrouwbare kennisbasis

Kernprincipe: kleine, genummerde bouwstenen plus citaatplicht.

Concreet:

- **Zet het Proceshuis om naar markdown-bestanden per onderwerp**, in vier mappen: `sales/`, `werven-selecteren/`, `ontwikkelen/`, `plaatsen/`. Eén bestand per

deelonderwerp (bijvoorbeeld `plaatsen/fase-a-contracten.md`, `ontwikkelen/doorzaam-budget.md`). Richtlijn: elk bestand behandelt één vraag die een medewerker echt stelt.

- **Elk bestand krijgt een kop met metadata:** eigenaar, laatst gecheckt op (datum), bron. Verouderde kennis is jullie grootste sluipende risico; een datum maakt dat zichtbaar.
- **De 15 kernregels gaan hard in de system prompt** van elke bot, niet alleen in de kennisbasis. Regels als "geen onboarding zonder kredietcheck" en "vrijwaring nooit tekenen" mogen nooit afhangen van of de zoekfunctie het juiste stuk vindt.
- **Citaatplicht in de prompt:** de bot mag alleen antwoorden met verwijzing naar het bronbestand ("volgens plaatsen/offboarding.md geldt..."). Vindt hij geen bron, dan is het verplichte antwoord: "Dit staat niet in het Proceshuis. Check bij [eigenaar van dit proces]." Dat is je "zegt wat hij niet weet"-mechanisme, en het is een promptregel, geen hoop.
- **Bouw een testset vóór je vertrouwt.** Laat de eerste gebruiker en een ervaren collega samen 30 vragen opschrijven: 20 waarvan het antwoord in het Proceshuis staat (met het juiste antwoord erbij) en 10 strikvragen waarvan het antwoord er níet in staat. De bot slaagt pas als hij de 20 goed beantwoordt mét bron, en bij alle 10 eerlijk "weet ik niet" zegt. Draai deze test opnieuw na elke grote wijziging.
- **Getallen zijn heilig:** omrekenfactor 2,53, de 2.500 euro Doorzaam, de 6-wekentermijn, de dag 3-10-17-cyclus. Zet deze in een apart bestand `kerngetallen.md` en test er expliciet op. Dit zijn de fouten die vertrouwen in één klap slopen.

---

c. Persoonlijk profiel per medewerker

Kernprincipe: de medewerker bouwt zijn eigen profiel mee. Dat is óók jullie verandermanagement.

Uit het gesprek: mensen schieten in de weerstand als iets vóelt als controle. Draai het om: het profiel is van de medewerker, niet over de medewerker.

Concreet:

- **`profiel.md` per medewerker, twee lagen** (precies zoals jullie guardrail 5 zegt):
  1. GreenFox-laag (voor iedereen gelijk, leidend): tone of voice, geen streepjes, niets dikgedrukt, geen emoji's, altijd terugkoppelen, servicebeloften.

2. Persoonlijke laag: rol en verantwoordelijkheden, vaste weektaken, schrijfstijl, dagritme (wanneer wil je je briefing), en wat de bot juist níet moet doen.

- **Vul het profiel via een startgesprek van een uur.** Hamza interviewt de medewerker: wat doe je op een normale dag, wat kost je de meeste tijd, welke drie mails schrijf je het vaakst. Plus: de medewerker levert 5 tot 10 zelfgeschreven mails aan als stijlvoorbeeld. Die voorbeelden zijn goud voor "herkenbaar persoonlijk, geen AI-mail".
- **De medewerker keurt zijn eigen profiel goed** en mag het altijd laten aanpassen (via Hamza). Eigenaarschap is het verschil tussen "mijn assistent" en "hun controlesysteem".
- **Harde afspraak, ook opschrijven:** de bot rapporteert nooit over de medewerker aan leidinggevenden. Geen gebruiksstatistieken per persoon naar management in fase 1. Anders bevestig je precies de angst die in jullie gesprek langskwam.
- **Laat het profiel meegroeien:** als de bot merkt dat de medewerker concepten steeds op dezelfde manier aanpast, stelt hij een profielwijziging voor. Medewerker zegt ja of nee, Hamza voert door. Zo wordt de bot vanzelf persoonlijker zonder dat het eng wordt.

---

d. Bouwvolgorde voor het snelst bewezen waarde

Kernprincipe: begin met het simpelste dat elke dag zichtbaar is. Koppel systemen pas als vertrouwen er is.

Volgorde voor gebruiker 1 (administratief-operationeel):

1. **Week 1-2: kennisbot (vraag-en-antwoord op het Proceshuis).** Geen koppelingen, geen data, alleen kennis met citaatplicht. Test met de 30-vragen-set. Dit is de fundering en het laagste risico.
2. **Week 2-3: de ochtendbriefing.** Elke ochtend een kort overzicht van openstaande acties. Input in het begin gewoon een handmatige lijst of export die de medewerker zelf bijhoudt. Waarom dit eerst: het is dagelijks zichtbaar, het went de medewerker aan de bot, en het raakt geen gevoelige data. Dit is jullie snelste route naar "ik gebruik hem uit mezelf".
3. **Week 3-4: conceptmails in eigen stijl.** Start met de drie meest voorkomende mailsoorten uit het profielgesprek. Medewerker beoordeelt elk concept met goed / bijna / fout, Hamza stuurt de prompt bij op basis daarvan.

4. **Week 4-6: bewaking van contracten en subsidieaanvragen.** Dit als laatste, want het is het foutgevoeligst (termijnen, de fase A-regel, de 6-wekentermijn) en vraagt om betrouwbare data-aanlevering. Begin met een wekelijkse export die de bot omzet naar een signaallijst ("deze 3 lopen binnen 2 weken af"), niet met een live koppeling.

Daarna gebruiker 2 (commercieel-relationeel): **hergebruik alles**, alleen een nieuw `profiel.md` en een startgesprek. Als dat meer kost dan een dag of twee, is de architectuur niet goed en wil je dat nú weten, niet bij medewerker nummer 12.

Meet vanaf dag één het succescriterium: gebruikt de medewerker de bot na twee weken uit zichzelf, en kan hij tijdwinst benoemen in uren per week. Laat Hamza dat wekelijks in één zin noteren.

En bewust niet doen: fase 2 (realtime meeluisteren via Plaud) pas starten als beide gebruikers fase 1 aantoonbaar dagelijks gebruiken. Meeluisteren raakt bovendien direct kandidaatdata en het gevoel van gecontroleerd worden. Dat verdien je eerst met een bot die simpelweg elke dag handig is.

Concrete eerste vervolgstap (deze week, zelf uitvoerbaar)

Hamza zet de fundering neer, nog vóór er één regel bot-logica staat:

1. Maak op de server de mappenstructuur: `kennis/` (met de vier proces-mappen plus `kerngetallen.md`) en `medewerkers/gebruiker-1/` (met leeg `profiel.md`, `taken.md`, `concepten/`). Zet alles onder git.
2. Zet de hoofdstukken Plaatsen en Ontwikkelen uit het Proceshuis om naar losse markdown-bestanden volgens het recept bij vraag b (één onderwerp per bestand, metadata-kop erboven).
3. Laat de eerste gebruiker samen met een ervaren collega de testset maken: 20 vragen mét het juiste antwoord en bronverwijzing, plus 10 vragen waarvan het antwoord bewust níet in het Proceshuis staat.

Leg het resultaat (mappenstructuur plus testset) volgende week aan mij voor. Dan review ik de structuur en geef ik het promptrecept voor de kennisbot met citaatplicht, zodat jullie die er direct bovenop kunnen bouwen.