

🔊 LUISTER DIT VERSLAG

-10%

1,0x

+10%

+25%

1,0x

↓ Download naar telefoon

▶ 📖 Lees mee met de podcast

545 berichten. 66 sessies. 14 juli tot 13 augustus 2026.

Wat werkt goed

- **Claude draait mee in de vloot.** Hij pakt taken uit de wachtrij van Paco, Codex en Jarvis, voert ze zelf uit en meldt terug.
 - **Altijd bewijs erbij.** HTTP 200-checks, exit-codes, bericht-ID's, bestandspaden. Geen loze "klaar".
 - **Ook een NEE wordt bewezen.** Bij een Chrome-policy kwam er geen blinde fix, maar drie losse checks die aantoonde dat de policy nooit bestond.
 - **Machines als één geheel.** Een gBrain/Postgres-storing van 14 uur opgespoord, plus de valse "alles groen"-melding die hem verborg.
 - **Zware klussen in één keer af.** De /insights-rapporten kwamen meestal in één ronde goed.
-

Wat je tegenhoudt

Bij Claude:

- **Schade als bijvangst.** 226 oude podcasts gearchiveerd → alle bestaande links stilletjes stuk.
- **Te ruime SQL.** Een LIKE-zoekopdracht pakte veel te veel berichten mee.
- **Oud proces dat nog schrijft.** Bijna een mp3-batch verpest.
- **"Dat kan ik niet"** — terwijl hij de tool er al voor gebouwd had.

Bij jou:

- **Te vroeg onderbreken.** Meerdere security-checks leverden niks op omdat je ingreep terwijl Claude nog aan het lezen was.
 - **Blokkades komen te laat boven.** Geen GUI, een lege API-sleutel, bestanden op een andere machine. Een check vooraf had dat in seconden laten zien.
-

Snelle winst

1. **PID-lockfile in elk batch-script.** Plus een pgrep die zichzelf uitsluit. Twee incidenten kwamen van precies dit gat: de podcast-batch en een wachtlus die een uur bleef hangen.
 2. **Eén FLEET.md die leeft.** Claude leest hem bij de start, werkt hem bij na elke wijziging. Dan hoeft hij niet steeds opnieuw uit te zoeken welke machine wat draait.
 3. **Altijd eerst een proefronde.** Bij archiveren, SQL-updates en batch-generatie: vraag eerst een lijst van wat er geraakt wordt. Dit ene gewoontetje had bijna alle zelf-veroorzaakte schade voorkomen.
-

Op termijn

- **Zelf-controlerende oplevering.** Elke wachtrij-taak sluit af met: links nalopen, elk bestand op HTTP 200 checken, sitemap vóór en ná vergelijken. Pas als dat slaagt mag "klaar" verstuurd worden.
- **Achterstand parallel wegwerken.** Nu serieel. Straks één sub-agent per los item, elk in een eigen worktree, bewijs naar één gedeelde tabel. Drie uur werk wordt dan één verzamelrapport.
- **Een machine-leesbaar capaciteitsregister.** Bij elke sessiestart verversen: welke machines bereikbaar zijn, welke sleutels nog saldo hebben, welk model er echt draait. Dan verdwijnt "ik kan dat niet" helemaal.