



VOOR DE IT-DIRECTIE EN TEAMLEADS

Agentic coding voor IT-teams

Jullie ontwikkelaars zetten coding agents in met een vaste werkwijze, in Claude Code of OpenAI Codex.

Turnleaf · Paul Van Cotthem · turnleaf.be

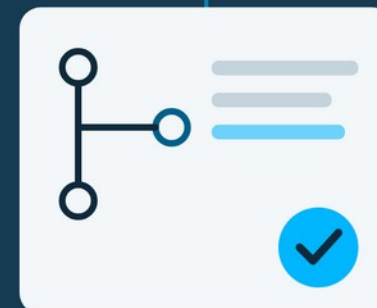
Ticket of
vraag



Agent volgt
jullie
standaarden



Developer
reviewt



HERKEN JE DIT?

Meer vraag naar software, zelfde team

De backlog groeit sneller dan het team

Vragen van de business, bugs en onderhoud: er blijft altijd meer liggen dan je kan oppakken.

Ontwikkelaars zijn schaars

Softwareontwikkelaar, tester en ICT-architect staan op de knelpuntlijst. Aanwerven duurt maanden.

Legacy en technische schuld remmen af

Oude systemen die weinig mensen nog kennen, upgrades die telkens worden uitgesteld.

AI wordt al gebruikt, maar los

Elke ontwikkelaar zijn eigen tool en account, zonder afspraken over rechten, data of review.

WAAROM NU

Schaarse profielen, weinig tijd om te bouwen

9%

van de Belgische ondernemingen
had moeilijk in te vullen vacatures
voor ICT-specialisten, tegenover
5% in de EU

Statistiek Vlaanderen op basis van
Eurostat, cijfers 2024

13

ICT-beroepen op de knelpuntlijst
van VDAB, waaronder
softwareontwikkelaar, tester en
ICT-architect

VDAB, Knelpuntberoepen 2026

16%

van hun tijd besteden
ontwikkelaars aan coderen; de rest
gaat naar zoeken, afstemmen en
wisselen van context

Atlassian State of DevEx 2025, 3.500
ontwikkelaars en managers in zes landen

84%

van de ontwikkelaars gebruikt AI-
tools of is het van plan, maar
slechts 33% vertrouwt de output

Stack Overflow Developer Survey 2025, ca.
49.000 respondenten, wereldwijd

Vier manieren om meer software te leveren

Optie	Voordeel	Nadeel
Extra ontwikkelaars aanwerven	Kennis in huis, volledige controle	Schaarse profielen, duur, maanden wervingstijd
Uitbesteden aan een IT-partner	Snel extra capaciteit, gespecialiseerd	Hoge kost, kennis van jullie code blijft buiten de deur
Een low-code platform	Snel voor eenvoudige toepassingen	Beperkt, nieuwe afhankelijkheid naast de bestaande code
Jullie ontwikkelaars werken met coding agents, met een vaste werkwijze	Meer doorvoer met hetzelfde team, kennis blijft in huis	Vraagt een opstart: dat is deze masterclass en dit traject

HET DOEL

Ontwikkelaars bepalen wat er gebouwd wordt. Agents schrijven en testen de code, volgens jullie eigen standaarden.

Niet: elke developer zijn eigen AI-tool. Wel: één werkwijze met afspraken in de repo, duidelijke rechten en altijd een menselijke review.

Vijf begrippen om te kennen

Projectinstructies

Afspraken over de repo

Een bestand dat de agent altijd leest: stack, conventies, hoe je test.

AGENTS.md of CLAUDE.md

Skill

Een vaste werkwijze

Een geschreven instructie voor een terugkerende taak.

Bv. jullie manier van migreren

Connector

Toegang tot systemen

Een beveiligde koppeling via MCP, met de rechten van de gebruiker.

Bv. Jira, Azure DevOps, Sentry

Rechten en sandbox

Wat de agent mag

Wat de agent mag lezen, schrijven en uitvoeren, en waar.

Bv. geen netwerk, geen productie

Achtergrondagent

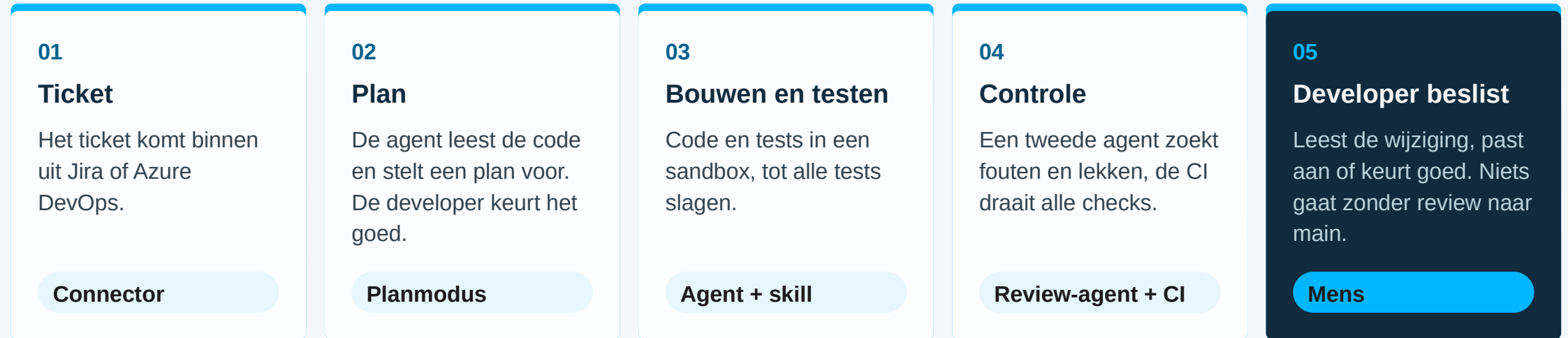
Werk dat doorloopt

De agent werkt zelfstandig in de cloud of in de CI-pijplijn.

Bv. CI-fouten op een PR herstellen

ZO ZIET EEN WORKFLOW ERUIT

Van ticket tot pull request



Dit werkt met Claude Code en met Codex: in de terminal, in de IDE of als achtergrondagent op GitHub.

WAT JE KAN AUTOMATISEREN

Acht workflows, van vandaag tot morgen

Workflow	Wat de agent doet	Status
Codebase verkennen	Legt architectuur en afhankelijkheden uit, schrijft documentatie	Vandaag
Tests schrijven	Zoekt ongeteste code, schrijft tests en draait ze tot ze slagen	Vandaag
Interne tools en scripts	Bouwt beheerschermen, rapporten en scripts vanuit een specificatie	Vandaag
Upgrades en refactoring	Zet code om naar een nieuwe versie, draait na elke stap de tests	Vandaag
Bug van ticket tot PR	Leest het ticket, reproduceert, herstelt en opent een pull request	Met koppeling
Eerste code review	Controleert wijzigingen op fouten en beveiliging, met commentaar	Met koppeling
Kwetsbare dependencies	Leest de scanresultaten, past versies aan, herstelt wat breekt	Met koppeling
Legacy moderniseren	Haalt bedrijfsregels uit oude code en herschrijft stap voor stap	In opbouw

Vandaag: op de repo en lokale tests · Met koppeling: ticketing, Git-platform of CI · In opbouw: vraagt een begeleide aanpak

Drie scenario's, één aanpak

SCENARIO A

Achterstand wegwerken

Bugs, tests en kleine wijzigingen lopen via agents. Ontwikkelaars reviewen en focussen op architectuur.

SCENARIO B

Legacy moderniseren

Oude code wordt gedocumenteerd, bedrijfsregels komen terug als specificatie, migratie in kleine stappen.

SCENARIO C

Veilig opschalen

AI wordt al gebruikt. Eén kader met gedeelde standaarden, rechten en review voor alle teams.

In alle drie bouwt het team dezelfde bouwstenen. Het verschil zit in waar jullie beginnen.

SCENARIO A

Achterstand wegwerken

Bugs, tests en kleine wijzigingen nemen de tijd in die nodig is voor architectuur en nieuwe projecten.

Vandaag

- bugs blijven weken liggen
- testdekking blijft laag
- tijd verloren aan zoeken en context
- reviews stapelen zich op

Met agents

- agent reproduceert, test en herstelt
- ontbrekende tests per module
- vragen over de code in seconden
- eerste reviewronde door een agent

Wat het oplevert

- kortere doorlooptijd van ticket tot PR
- hogere testdekking
- ontwikkelaars op het moeilijke werk
- meetbaar met de DORA-meetpunten

SCENARIO B

Legacy moderniseren

Kennis van het oude systeem zit bij een paar mensen. Een migratie lijkt een project van jaren.

Vandaag

- documentatie ontbreekt of is verouderd
- afhankelijk van enkele collega's
- upgrades worden uitgesteld
- angst voor regressies

Met agents

- agent leest de code en documenteert
- bedrijfsregels terug als specificatie
- eerst tests, dan migreren per module
- elke stap via PR en review

Wat het oplevert

- kennis vastgelegd in de repo
- kleine, veilige stappen
- minder afhankelijk van enkelingen
- de migratie komt in beweging

SCENARIO C

Veilig opschalen

Ontwikkelaars gebruiken al AI, elk op hun manier. Security en architectuur horen het achteraf.

Vandaag

- eigen tools en privé-accounts
- geen afspraken over rechten of data
- geen zicht op gebruik of kosten
- kwaliteit verschilt per persoon

Met een kader

- zakelijke accounts, centraal beheerd
- gedeelde projectinstructies en skills
- sandbox en goedgekeurde connectors
- verplichte review en CI-checks

Wat het oplevert

- één werkwijze in alle teams
- zicht op gebruik en kosten
- beleid dat je kan aantonen
- snellere onboarding van nieuwe collega's

IN ELK SCENARIO

Alle voordelen op een rij



Kortere doorlooptijd

Van ticket tot pull request in uren in plaats van dagen.



Meer tests

Testdekking die eindelijk omhoog gaat.



Legacy in beweging

Oude code gedocumenteerd en stap voor stap vernieuwd.



Snellere onboarding

Nieuwe collega's laten de agent de code uitleggen.



Gedeelde standaarden

Afspraken in de repo, niet in hoofden.



Minder sleurwerk

Ontwikkelaars op architectuur en moeilijke problemen.



Zicht en controle

Gebruik, kosten en kwaliteit per team opvolgen.



AI-geletterdheid

Een concrete maatregel onder artikel 4 van de AI-verordening.

Twée gelijkwaardige tools, dezelfde aanpak

Optie 1 · Anthropic

Claude Code

Coding agent in terminal, VS Code, JetBrains, desktopapp en browser.

- projectinstructies in CLAUDE.md
- skills, subagents, hooks, plugins en MCP
- zes rechtenmodi en een sandbox
- code review op GitHub (preview)

Optie 2 · OpenAI

OpenAI Codex

Coding agent met open-source CLI, IDE, ChatGPT-desktopapp en Codex Cloud.

- projectinstructies in AGENTS.md
- skills, plugins en MCP
- sandbox, standaard zonder netwerk
- code review op GitHub en in de app



Open standaarden: AGENTS.md, MCP en Agent Skills werken in beide tools. Werken jullie met GitHub? Dan kunnen beide agents ook binnen GitHub Copilot draaien (preview).

Resultaten bij andere organisaties

2 uur

in plaats van 2 weken voor
een prototype; ca. 700
gebruikers, uitgerold met
praktijkworkshops

Pictet (CH), private bank, met Claude
Code

2-3 dagen

in plaats van 2-3 weken voor
sommige projecten; ca. 1.000
medewerkers bouwen ermee

AutoScout24 (DE), met Codex, 2026

30-60 min

in plaats van 2 weken voor
een refactoring van legacy-
code; geen P1-incidenten bij
de lancering van de app

Virgin Atlantic (UK), met Codex,
22/05/2026

99,1%

van 2.913 API's gemigreerd
uit een systeem van 20 jaar
oud, in 7 maanden

LG CNS (Zuid-Korea), met Claude
Code

WAT ONAFHANKELIJK ONDERZOEK ZEGT

De winst zit in de werkwijze, niet in de tool

-19%

trager werkten ervaren ontwikkelaars met AI in 2025, terwijl ze dachten 20% sneller te zijn

METR, gerandomiseerde studie, 16 ontwikkelaars; vervolg 2026: zwak bewijs van winst

45%

van de AI-gegenereerde code bevatte een beveiligingslek, over 80 taken en meer dan 100 modellen

Veracode GenAI Code Security Report, 2025

8x

meer dubbele codeblokken in 2024; minder refactoring (verplaatste code -39,9%)

GitClear, analyse van 211 miljoen regels code, 2025

90%

gebruikt AI; meer doorvoer, maar minder stabiele releases

DORA 2025 (Google), ca. 5.000 respondenten

De risico's, en hoe we ze beheersen

Risico	Wat we afspreken
Kwaliteit en veiligheid	Elke wijziging via pull request, tests en menselijke review, plus een beveiligingscheck in de CI.
Rechten van de agent	Sandbox, geen productiegeheimen, enkel goedgekeurde connectors. Tekst in code of tickets kan een agent misleiden.
Broncode en data	Zakelijke accounts, geen training op jullie code. Verwerking in de EU kan bij beide tools, met voorwaarden.
Verzonnen packages	Agents noemen soms packages die niet bestaan en die aanvallers registreren. Enkel goedgekeurde bronnen, scans in de CI.
Vaardigheden van junioren	In een studie scoorden junioren met AI 50% op begrip van nieuwe code, zonder AI 67%. We bouwen leermomenten in.
AI-verordening	Artikel 4 vraagt maatregelen voor AI-geletterdheid. Deze masterclass is zo'n maatregel.

Zo blijft de factuur voorspelbaar

Wat het verbruik doet stijgen

- lange sessies: elke stap stuurt de hele voorgeschiedenis opnieuw mee
- vage opdrachten: de agent zoekt en probeert meer
- het zwaarste model voor elke taak
- veel agents en koppelingen tegelijk actief

Opvolgen: meet eerst een basislijn in een pilot en volg de kost per afgewerkte taak. Beide tools bieden limieten en rapporten per gebruiker.



Juiste model per taak

Het sterkste model voor architectuur, een lichter model voor routinewerk.



Korte, gerichte sessies

Eén taak per sessie, daarna opnieuw beginnen of samenvatten.



Lichte instructies

Een kort basisbestand; vakkennis in skills die pas laden als ze nodig zijn.



Budget per team

Limieten per gebruiker of team, opgevolgd in het beheerdashboard.

De instap: een werksessie op jullie eigen code

-  Werksessie van 6 uur, bij jullie op locatie
-  Maximaal vijf ontwikkelaars per sessie
-  Op jullie eigen repo's, toestellen en accounts
-  Claude Code of OpenAI Codex, naar keuze
-  Syllabus als pdf, met alle updates

Jullie gaan naar huis met

- projectinstructies voor een eigen repo
- een eerste gedeelde skill
- een workflow van ticket tot pull request
- afspraken over rechten en review
- zicht op wat een taak verbruikt

HET TRAJECT

Van masterclass tot vaste werkwijze

01

Masterclass

Het team leert werken met agents op de eigen code, met de tool van jullie keuze.

Eerste workflows

02

Spelregels

Rechten, sandbox, connectors, data-locatie en budgetten afstemmen met security en architectuur.

Kader

03

Pilot

Gedeelde skills en projectinstructies, koppeling met ticketing en CI, in één team.

Werkende pilot

04

Uitrollen

Meer teams, resultaten meten, coaching en bijsturen.

Vaste werkwijze

Inhoud, tempo en duur van elke fase stemmen we af na een eerste gesprek. Je krijgt daarna een voorstel op maat.

RESULTAAT

Wat jullie overhouden

- ✓ **Werkende workflows** op jullie eigen codebase.
- ✓ **Projectinstructies en skills** in de repo, gedeeld door het team.
- ✓ **Spelregels** voor rechten, review en data, afgestemd met security.
- ✓ **Koppelingen** met ticketing en CI waar dat loont.
- ✓ **Meetpunten** om het resultaat aan de directie te tonen.
- ✓ **De syllabus als pdf**, met updates.

Voor wie?

IT-directie, teamleads, ontwikkelaars, testers en DevOps.

Voorkennis

Ervaring met programmeren en Git. Basiskennis van een AI-assistent volstaat; de agent leer je in de sessie.

VOLGENDE STAP

Een gratis verkennend gesprek

Daarin bespreken we:

- welke repo's en taken nu de meeste tijd kosten
- welke tools en accounts IT toelaat: Claude, OpenAI of GitHub Copilot
- het team en de eerste workflow voor de pilot

CONTACT

Paul Van Cotthem

paul@turnleaf.be

+32 499 77 77 77

turnleaf.be

Meer dan 25 jaar in commerciële en strategische rollen bij Telenet, Apple en Xerox. Sinds 2023 meer dan 240 organisaties begeleid.



TECHNISCHE BIJLAGE

Voor teamleads en architecten: de twee tools naast elkaar, en de spelregels.

Stand van zaken op 08/10/2026. Deze tools veranderen snel; we werken de vergelijking bij voor elke sessie.

Claude Code en Codex naast elkaar: werking

	Claude Code	OpenAI Codex
Waar het draait	Terminal, VS Code, JetBrains (bèta), desktopapp, browser, GitHub Actions	Open-source CLI, VS Code, JetBrains, ChatGPT-desktopapp, browser, Codex Cloud
Projectinstructies	CLAUDE.md; leest AGENTS.md als er geen CLAUDE.md is	AGENTS.md, ook per map in de repo
Uitbreiden	Skills, subagents, hooks, plugins, MCP	Skills, plugins met hooks, MCP
Rechten en sandbox	Zes rechtenmodi, waaronder een automatische modus met controlemodel; sandbox met netwerk-allowlist	Drie sandboxmodi plus goedkeuringsregels; netwerk standaard uit
Achtergrondagents	Cloudsessies, geplande runs, CI-fouten automatisch herstellen	Codex Cloud; bij Enterprise standaard uitgeschakeld

BIJLAGE

Claude Code en Codex naast elkaar: organisatie

	Claude Code	OpenAI Codex
Code review op GitHub	Research preview, voor Team en Enterprise	Beschikbaar op GitHub; GitLab in preview
Beheer	Beheerde instellingen, SSO, SCIM, Compliance API, analytics	Rollen, beheerde configuratie, Compliance API, Analytics API
Training op jullie data	Niet onder zakelijke voorwaarden	Niet standaard bij Business, Enterprise en API
Verwerking in de EU	Niet bij Anthropic zelf; wel via AWS Bedrock en Google Cloud	API en nieuwe Enterprise-klanten; Codex-app en CLI wel, Codex Web niet
Modellen	Claude Fable 5.1, Opus 5.5, Sonnet 5.5, Haiku 5.5	GPT-6.1 Sol, GPT-6 Luna, GPT-6 Astra
Prijmodel	Licenties in Team of Enterprise, of per gebruik via API	In ChatGPT-abonnementen met credits, of per gebruik via API

Spelregels die we samen vastleggen

- ✓ Enkel zakelijke accounts, beheerd door IT
- ✓ Projectinstructies (AGENTS.md) in elke repo
- ✓ Sandbox aan, geen productiegeheimen in de omgeving
- ✓ Een lijst van goedgekeurde connectors

- ✓ Branchbeveiliging: geen merge zonder menselijke review
- ✓ Scans op dependencies en geheimen in de CI
- ✓ Budgetlimieten per team, kost per taak opvolgen
- ✓ Meetpunten: doorlooptijd, faalpercentage, hersteltijd