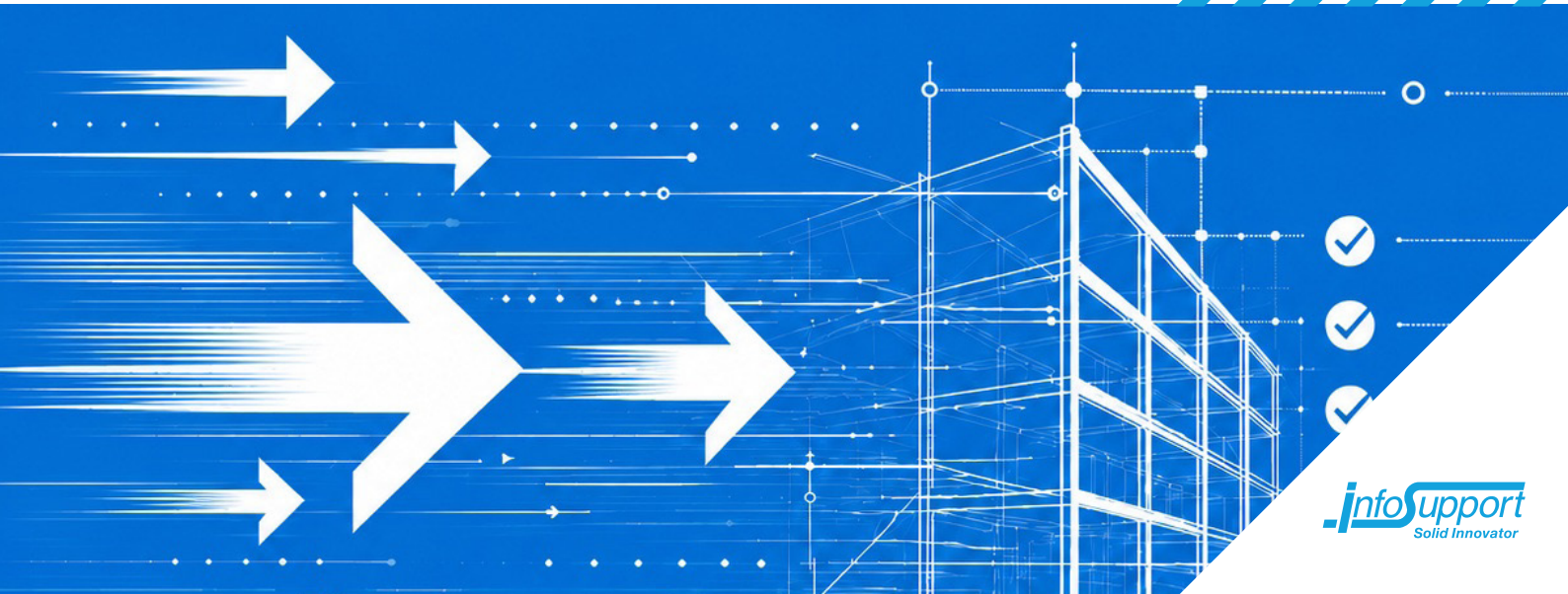
Whitepaper

2026

Sneller bouwen vraagt om sterker vakmanschap



Solid Innovator

Voorwoord

AI wordt binnen Nederlandse organisaties al breed ingezet voor softwareontwikkeling, maar de stap naar beheersbare agentic engineering blijft achter. Ons Agentic Engineering-onderzoek 2026, uitgevoerd in samenwerking met Markteffect onder 330 Nederlandse IT-beslissers*, laat zien dat **84 procent** van de organisaties AI inzet tijdens het ontwikkelproces. Tegelijkertijd bevindt slechts **16 procent** zich in de integratie- of optimalisatiefase, is bij **8 procent** AI-governance formeel verankerd en houdt slechts **31 procent** actief toezicht op AI-agents.

Die cijfers laten zien dat AI het schrijven, testen en controleren van code inmiddels versnelt, terwijl de organisatie eromheen nog niet is meegegroeid. De bottleneck verschuift daardoor naar requirements, architectuur, deployment en beheer. Teams produceren sneller code, maar houden die snelheid niet vast zodra ze software moeten specificeren, beoordelen, integreren en beheren.

In dit whitepaper delen we de belangrijkste resultaten van het onderzoek. We laten zien waar organisaties vandaag staan, waar hoofdbeslissers en adviseurs verschillend naar dezelfde situatie kijken en welke randvoorwaarden nodig zijn om agentic engineering beheersbaar op te schalen. We vullen de cijfers aan met de visie van onze experts, zodat je de stap van individueel gebruik naar een volwassen engineeringpraktijk gericht kunt zetten.

**Het Agentic Engineering-onderzoek 2026 is uitgevoerd in samenwerking met Markteffect onder 330 IT-beslissers. De volledige onderzoeksverantwoording staat achterin dit whitepaper.*

Joop Snijder
Head of AI

Frank Thiele
Unit Manager

Bas Meerman
Managing Director

Inhoudsopgave

1. Van AI-gebruik naar agentic engineering 06

2. Eén organisatie, verschillende perspectieven 10

3. Betrouwbaarder bouwen vraagt om een sterker AI coding harness 14

4. De engineer verdwijnt niet, maar het vak verandert wel 20

5. Op hoeveel gehuurde grond wil je bouwen? 25

6. Van pilot naar schaalbare engineeringpraktijk 30

Conclusie 35

Samenvatting

AI wordt al breed gebruikt bij softwareontwikkeling binnen Nederlandse organisaties, maar de stap naar beheersbare agentic engineering is door veel bedrijven nog niet gezet. Uit het Agentic Engineering-onderzoek 2026, dat Info Support liet uitvoeren onder 330 Nederlandse IT-beslissers, blijkt dat 84 procent van de organisaties AI inzet tijdens het ontwikkelproces. Maar er is nog geen sprake van een volwassen toepassing. Slechts 34 procent noemt het gebruik van AI standaard, 16 procent bevindt zich in de integratie- of optimalisatiefase en bij slechts 8 procent is AI-governance formeel verankerd met verantwoordelijkheden en een roadmap. Ook toezicht op AI-agents is nog niet vanzelfsprekend: bij 31 procent vindt actief toezicht plaats via logging, audits en een duidelijke rolverdeling.

AI versnelt softwareontwikkeling, maar organisaties worden daardoor niet automatisch sneller. Het onderzoek laat zien dat AI vandaag vooral coding, code review en testing versnelt. Daardoor verschuift de bottleneck naar andere onderdelen van het ontwikkelproces, zoals requirements, architectuur, deployment en beheer.

Teams produceren sneller code, maar hebben moeite om diezelfde snelheid vast te houden zodra ze software moeten specificeren, beoordelen, integreren en beheren. AI versnelt softwareontwikkeling, maar organisaties worden daardoor niet automatisch sneller. Het onderzoek laat zien dat AI vandaag vooral coding, code review en testing versnelt. Daardoor verschuift de bottleneck naar andere onderdelen van het ontwikkelproces, zoals requirements, architectuur, deployment en beheer. Teams produceren sneller code, maar hebben moeite om diezelfde snelheid vast te houden zodra ze software moeten specificeren, beoordelen, integreren en beheren.

Wij zien twee voorwaarden om duurzame waarde uit agentic engineering te halen. Ten eerste wordt vakmanschap belangrijker. Agents hebben geen vanzelfsprekend begrip van context, risico en nuance. Daarom hebben ze een duidelijk AI coding harness nodig: een omgeving waarin architectuurprincipes, specificaties, tests, securityregels en feedbackloops zijn vastgelegd én technisch worden afgedwongen.

Zo weet een agent wat hij mag doen, welke context hij krijgt en wanneer hij moet stoppen of om hulp moet vragen. Een AI coding harness gaat net zo vanzelfsprekend worden als CI/CD. Ten tweede moeten organisaties bewust nadenken over hun afhankelijkheid van modellen en platformleveranciers. Wie agentic engineering inzet voor bedrijfskritische software, moet ook keuzes maken over portabiliteit, hybride inzet en digitale soevereiniteit.

De volgende stap is om AI niet alleen door individuele developers te laten gebruiken, maar de werkwijze structureel in teams en processen te verankeren. Pas wanneer AI onderdeel wordt van hoe teams samenwerken, software ontwikkelen, kwaliteit bewaken en kennis delen, komt de werkelijke waarde van agentic engineering naar boven. Tegelijk groeit dan ook de vraag hoe organisaties die inzet beheersbaar houden. Een AI coding harness helpt om die stap gecontroleerd te zetten.

Opvallend aan de resultaten van het onderzoek is dat hoofdbeslissers en adviseurs anders naar dezelfde situatie kijken. Hoofdbeslissers zien volwassenheid, vertrouwen en verandering vaak optimistischer dan adviseurs.

Als die beelden te ver uit elkaar liggen, kunnen organisaties AI-agents meer ruimte geven dan hun processen, security en kwaliteitscontroles aankunnen. Dan komt niet alleen de kwaliteit van software onder druk te staan, maar ook de veiligheid en continuïteit van bedrijfskritische systemen. Dat maakt een gedeelde reality check een logische eerste stap. Dat betekent: eerst scherp krijgen waar AI vandaag wordt gebruikt, welke controles al goed functioneren, waar risico's zitten en welke vaardigheden nog ontbreken. Hoe organisaties dat kunnen aanpakken, lees je in de volgende hoofdstukken.

1. Van AI-gebruik naar agentic engineering

AI gebruiken is nog geen agentic engineering

Steeds meer organisaties gebruiken AI tijdens softwareontwikkeling. Dat is een belangrijke eerste stap, maar het is nog geen agentic engineering. Die verschuiving vindt plaats zodra AI meer wordt dan een persoonlijke assistent van de developer en een vaste plek krijgt in het softwareontwikkelproces. De focus ligt dan niet meer alleen op individuele productiviteit, maar op de vraag hoe mensen, AI-agents en engineeringprocessen betrouwbaar met elkaar kunnen samenwerken.

Wat betekent dit voor organisaties?

Deze stap vraagt niet om AI-tools simpelweg breder uit te rollen, maar om een andere engineeringaanpak. Teams hebben duidelijke werkwijzen nodig, kwaliteitskaders, geautomatiseerde bewaking van die kaders, passende infrastructuur en vaak ook nieuwe vaardigheden.

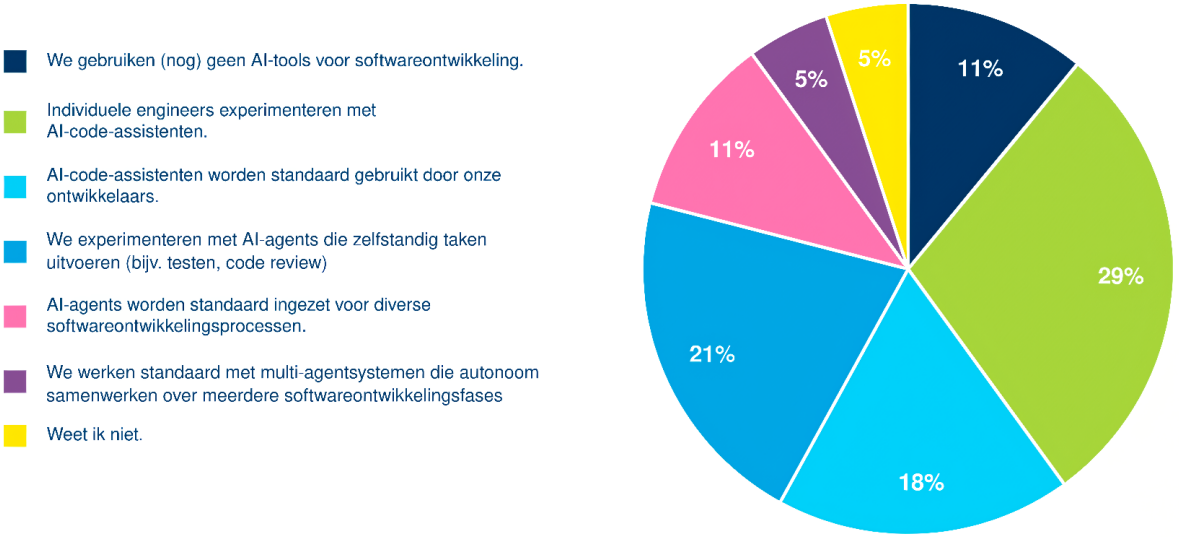
Het moet ook duidelijk zijn wat teams wel en niet met AI mogen doen, wie afspraken kan aanpassen en wanneer nieuwe risico's daarom vragen. Zodra AI onderdeel wordt van hoe teams samenwerken, software ontwikkelen, kwaliteit bewaken en kennis delen, levert die productiviteitswinst ook echt waarde op.

Wat laat het onderzoek zien?

AI krijgt al breed een plek in softwareontwikkeling, maar geïntegreerde en beheersbare agentic engineering staat bij veel organisaties nog in de kinderschoenen. Het onderzoek laat zien waar AI vandaag vooral wordt ingezet en dat de stap naar een volwassen agentic engineeringpraktijk nog groot is.

Figuur 1

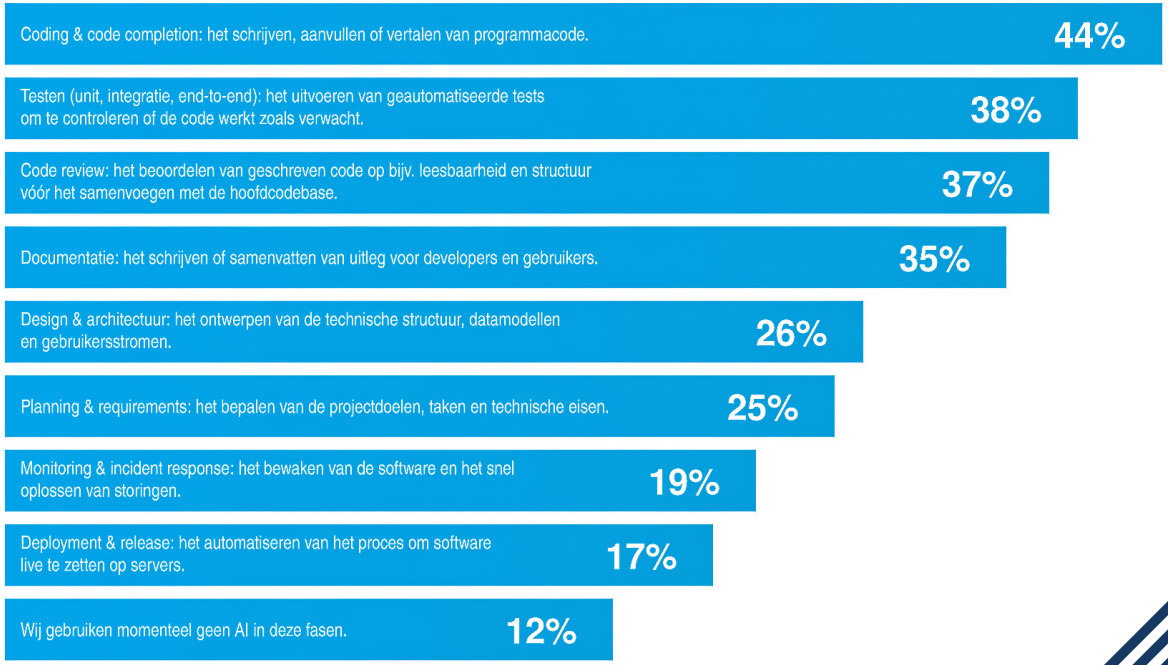
In welke mate wordt AI op dit moment gebruikt voor jullie softwareontwikkeling?



Figuur 2

Voor welke fasen van softwareontwikkeling zet jouw organisatie momenteel AI in?

Respondenten konden meerdere antwoorden aanvinken.



Het onderzoek laat een duidelijke trechter zien. Veel organisaties gebruiken AI voor softwareontwikkeling, maar bij elke stap richting volwassen toepassing wordt de groep kleiner. Zo geeft minimaal een derde (**34%**) aan dat AI standaard wordt gebruikt en bevindt **16 procent** zich in de integratie- of optimalisatiefase. Bij **8 procent** is AI-governance formeel verankerd met verantwoordelijkheden en een strategische roadmap.

Dat patroon zegt meer dan het hoge adoptiepercentage alleen. Een code-assistent is relatief snel ingevoerd, maar agentic engineering vraagt om meer samenhang. Teams moeten hun werkwijze, tooling, architectuur, kwaliteit, vaardigheden en leiderschap op elkaar afstemmen. Juist in die overgang van losse toepassing naar betrouwbare engineeringpraktijk laat de markt nog tekortkomingen zien.

Frank Thiele, Unit Manager, legt uit:

> *"AI helpt vandaag vooral bij het schrijven, testen en controleren van code. Dat levert snelheid op, maar softwareontwikkeling bestaat uit meer dan coding alleen. Als requirements, architectuur, acceptatie, deployment en beheer niet mee-evolueren, verschuift de bottle-neck gewoon verder in het proces. Dan produceren teams sneller output, maar niet automatisch sneller waarde. Organisaties die agentic engineering serieus willen inzetten, moeten daarom breder kijken dan codegeneratie alleen."*

Wat laat het onderzoek zien?

Code-assistenten helpen vandaag vooral individuele developers, maar AI-agents gaan een stap verder. Ze kunnen bijvoorbeeld een plan maken, code aanpassen, tests draaien, feedback verwerken en taken uitvoeren over meerdere fases van de softwarelevenscyclus. Daarmee verschuift ook de rol van engineers: zij schrijven weliswaar minder handmatig code, maar krijgen meer verantwoordelijkheid voor de context waarin agents werken. Wie agentic engineering doeltreffend wil inzetten, moet daarom niet vooral extra AI-tools toevoegen, maar het engineeringproces rond AI goed inrichten.

2. Eén organisatie, verschillende perspectieven

Eerst een gedeeld vertrekpunt

Succesvolle adoptie van agentic engineering begint met een gedeeld beeld van waar de organisatie staat, hoeveel autonomie wenselijk is en welke mate van controle nodig is. Toch kijken stakeholders daar niet altijd hetzelfde naar. Zo vindt 24 procent van de hoofdbeslissers dat elke AI-actie menselijke controle vereist, tegenover 50 procent van de adviseurs met invloed. Dat verschil is belangrijk, omdat het bepaalt hoeveel ruimte AI-agents krijgen, welke controles nodig zijn en hoe snel een organisatie verantwoord kan opschalen. Zonder zo'n gedeeld vertrekpunt kunnen de ambities al snel vooruitlopen op wat teams technisch en organisatorisch aankunnen.

Wat betekent dit voor organisaties?

Een reality check is daarom een logische eerste stap om samen scherp te krijgen waar waarde zit, waar risico's ontstaan en welke verbeteringen eerst nodig zijn. Als bestuur, architecten en ontwikkelteams vanuit hetzelfde vertrekpunt werken, kunnen organisaties beter bepalen waar ze AI-agents verantwoord kunnen opschalen, welke vaardigheden ontbreken en welke kwaliteitskaders of controles nodig zijn.

Wat laat het onderzoek zien?

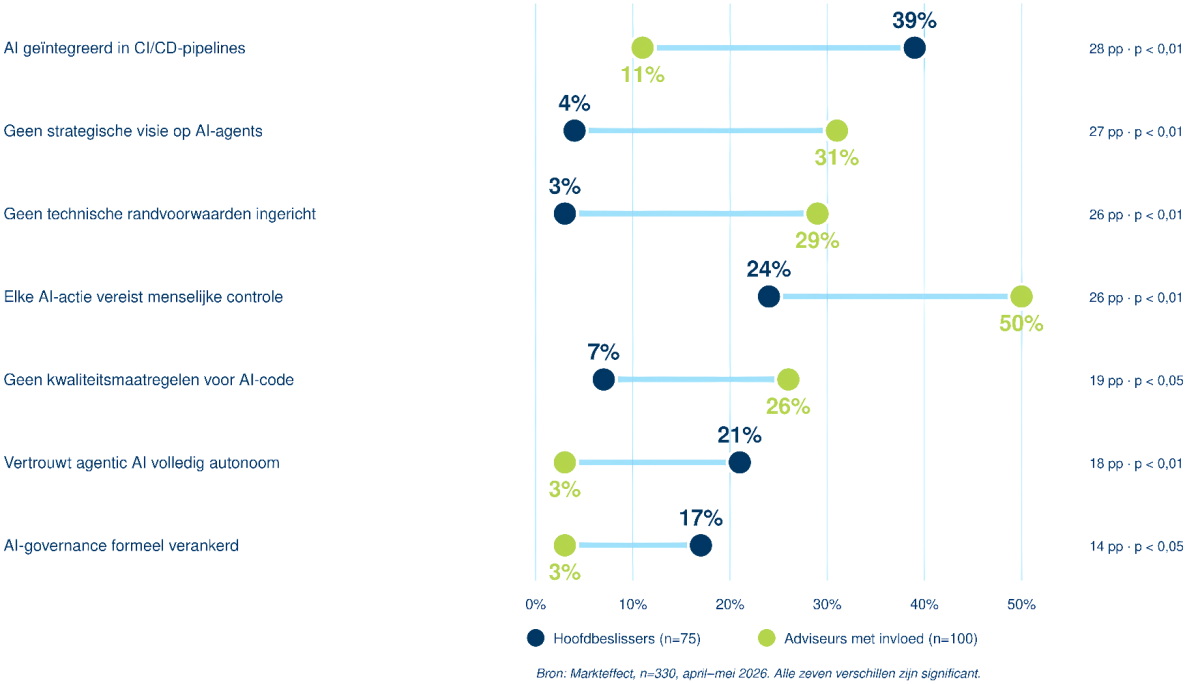
Het Agentic Engineering-onderzoek laat zien dat hoofdbeslissers, medebeslissers en adviseurs met invloed niet altijd hetzelfde beeld hebben van de stand van AI in de organisatie. Hoofdbeslissers schatten de volwassenheid, het vertrouwen en de controle vaker optimistisch in dan adviseurs.

Het Agentic Engineering-onderzoek laat zien dat hoofdbeslissers, medebeslissers en adviseurs met invloed niet altijd hetzelfde beeld hebben van de stand van AI in de organisatie. Hoofdbeslissers schatten de volwassenheid, het vertrouwen en de controle vaker optimistisch in dan adviseurs.

Figuur 3

Dezelfde organisaties, een ander beeld: beslissers rapporteren consequent rooskleuriger

Hoofdbeslissers en adviseurs met invloed kijken aantoonbaar anders naar dezelfde organisatie.



Bron: Marktеffect, n=330, april–mei 2026. Alle zeven verschillen zijn significant.

Joop Snijder
Head of AI

“Deze cijfers laten niet zien wie er gelijk heeft, maar wel dat de top en de werkvloer verschillend kijken naar hoe klaar de organisatie is voor opschaling. Als de top denkt dat de organisatie er klaar voor is, terwijl teams nog vragen hebben over kwaliteit, risico en eigenaarschap, ontstaan losse initiatieven en discussie over wat verantwoord is. Een gedeelde reality check helpt om die discussie concreet te maken: waar staan we echt, welke controles werken al en waar moeten we eerst bijsturen? Zo kunnen organisaties gerichter opschalen, met duidelijkere prioriteiten en meer vertrouwen tussen bestuur, architecten en teams.”

Reality check: vijf vragen voor organisaties

Onderstaande checklist helpt organisaties hun vertrekpunt concreet te maken. De uitkomst hoeft geen generieke maturityscore te zijn, maar wel een concreet beslissingsdocument.

1. Welke AI-toepassingen gebruiken teams vandaag al in hun dagelijkse werk?
2. Welke winst levert dat op, en waar blijft die uit?
3. Welke controles zijn er om AI-output te beoordelen en risico's te beperken?
4. Welke vaardigheden ontbreken nog in teams?
5. Waar mag een AI-agent zelfstandig handelen, en waar blijft menselijke controle nodig?

Wil je deze reality check niet alleen uitvoeren?

Onze AI-experts helpen organisaties om samen scherp te krijgen waar ze staan en waar de eerste verbeterstappen liggen.

3. Betrouwbaarder bouwen vraagt om een sterker AI coding harness

De omgeving bepaalt de kwaliteit

Een AI-model levert niet automatisch betrouwbare software op. De kwaliteit van software wordt steeds minder bepaald door het model alleen en steeds meer door de engineeringomgeving waarin AI-agents werken. Die omgeving bepaalt welke context AI-agents meekrijgen, welke regels ze moeten volgen en hoe hun werk wordt gecontroleerd en bijgestuurd.

Zonder die kaders kan AI snel veel output maken, maar blijft onduidelijk of die output veilig en onderhoudbaar is en aan de afgesproken kwaliteitscriteria voldoet.

Wat betekent dit voor organisaties?

De focus verschuift daarom van toolkeuze naar engineeringontwerp. Niet alleen het AI-model telt, maar vooral de werkomgeving waarin de agent taken uitvoert. Organisaties moeten dus bepalen hoe agents toegang krijgen tot context, hoe ze feedback ontvangen, welke acties ze mogen uitvoeren en wanneer ze moeten stoppen of om hulp vragen. Een AI coding harness brengt die werkomgeving samen.

Het bestaat uit de tools, feedbackloops, context en guardrails die bepalen wat een agent kan zien en doen. Denk aan toegang tot de juiste codebase, tests die de agent kan draaien, linters en type-checkers die fouten signaleren, een sandbox waarin code veilig draait en duidelijke afspraken over toegestane bestanden of commando's. Zo corrigeert het proces de agent continu en blijft zijn werk binnen veilige, controleerbare kaders.

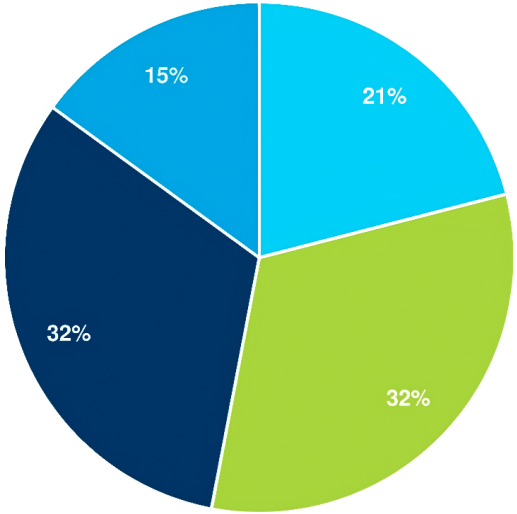
Wat laat het onderzoek zien?

Veel organisaties zetten AI al in, maar de manier waarop zij AI beheersbaar maken verschilt sterk. Slechts een minderheid heeft AI volledig geïntegreerd in het kwaliteitsproces, terwijl organisaties tegelijk investeren in maatregelen als goedkeuring van AI-tools, logging en sandboxomgevingen. Daarmee verschuift de aandacht van het AI-model naar de omgeving waarin AI werkt. Naarmate AI-agents zelfstandiger opereren, draait het niet alleen om codekwaliteit, maar ook om duidelijke kaders voor wat zij mogen doen, welke context zij gebruiken en hoe hun werk wordt gecontroleerd. Een AI coding harness brengt die technische en organisatorische guardrails samen.

Figuur 4

In hoeverre gelden jullie standaard kwaliteitseisen voor code die door AI is gemaakt?

- We gebruiken AI alleen voor snelle tests; deze code hoeft niet aan officiële standaarden te voldoen.
- We kijken de code wel na (code review), maar het doorloopt nog niet onze volledige test- en automatiseringsronde.
- AI-code moet aan precies dezelfde eisen voldoen als code van een programmeur.
- AI is volledig opgenomen in ons kwaliteitsproces: reviews, testen en CI/CD verlopen deels ook met AI-ondersteuning.



Vakmanschap wordt belangrijker naarmate AI meer overneemt

Hoe meer taken AI overneemt, hoe belangrijker softwarevakmanschap wordt. Juist doordat AI steeds meer code produceert, worden architectuur, specificatie, testautomatisering, security en kwaliteitskaders belangrijker. Niet omdat mensen elke stap handmatig moeten controleren, maar omdat organisaties AI pas veilig kunnen opschalen als kwaliteit structureel in het proces zit.

Veilig werken vraagt om technische kaders

Om kwaliteit en veiligheid binnen software-engineering te waarborgen, zetten organisaties allereerst in op security- en compliancemaatregelen voor data-uitwisseling met AI-tools (**42%**) en een goedkeuringsproces voor nieuwe AI-tools (**41%**). Logging en monitoring van AI-toolgebruik (**32%**) en aparte sandbox- of testomgevingen (**26%**) zijn andere maatregelen die worden ingezet.

Hoe ziet een AI coding harness eruit?

Onderstaand schema laat zien hoe een AI coding harness kan worden opgebouwd. Onze engineers helpen organisaties zo'n harness in te richten binnen bestaande pipelines en platformen.

De omgeving waarin een AI-agent werkt, bepaalt wat hij ziet, mag en oplevert.

Context	Tools	Guardrails	Feedback
• Toegang tot de juiste code-base • Specificaties en acceptatie-criteria • Architectuurprincipes en patronen	• Tests die de agent zelf kan draaien • Linters en type-checkers • Sandbox waarin code veilig draait	• Afspraken over toegestane bestanden • Toegestane commando's en rechten • Security- en compliance-kaders	• Signalering van fouten en afwijkingen • Reviews door engineers • Bijsturing tot het werk voldoet

De agent werkt binnen deze omgeving en stopt of vraagt om hulp zodra hij buiten de afgesproken kaders komt.

Drie lagen voor passende AI-kaders

Van een gedeelde professionele basis naar afspraken per toepassing.

	REIKWIJDTE	VOORBEELDEN	DOEL
LAAG 01 **Engineering-principes**	Herbruikbaar over organisaties heen	Coding standards, secure defaults, teststrategie, reviewcriteria	Een minimale professionele basis
LAAG 02 **Organisatie-kaders**	Specifiek voor de organisatie	Doelarchitectuur, technologiekeuzes, platformstandaarden, compliance	Consistentie en aansluiting op het IT-landschap
LAAG 03 **Applicatie-kaders**	Specifiek voor een oplossing en het proces	Domeinregels, risico's, performance-eisen, dataclassificatie, acceptatiecriteria	Passende autonomie en kwaliteit per toepassing

Frank Thiele
Unit Manager

> *“Een organisatie heeft niet één universeel harnas voor alle software. Een interne medewerkersapp vraagt om andere controle dan een systeem dat energie-infrastructuur aanstuurt of financiële transacties verwerkt. Daarom moet je gemeenschappelijke principes herbruikbaar maken, maar teams ook ruimte geven om per applicatie strengere of andere kaders toe te passen. Zo maak je governance praktisch: niet als beleidsdocument naast het ontwikkelproces, maar als uitvoerbare regels in je harnas, repositories, pipelines, agents, testomgevingen en platformvoorzieningen.”*

4. De engineer verdwijnt niet, maar het vak verandert wel

De egineer blijft aan zet

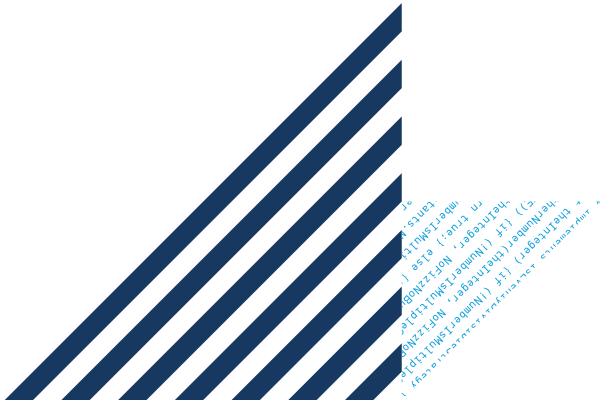
AI neemt steeds meer uitvoerende taken over, maar daarmee verdwijnt de software engineer niet uit beeld. Integendeel: hoe meer agents uitvoeren, hoe belangrijker de rol van engineers wordt. Zij geven richting aan de architectuur, bewaken de kwaliteit en bepalen wanneer AI-output goed genoeg is. De waarde van agentic engineering zit dus niet in het volledig automatiseren van softwareontwikkeling, maar in de samenwerking tussen AI en vakmanschap.

Wat betekent dit voor organisaties?

Investeren in vakmanschap wordt minstens zo belangrijk als investeren in tooling. Engineers moeten niet alleen leren werken met AI-agents, maar ook goede opdrachten formuleren, output kritisch beoordelen en risico's herkennen. Een ervaren engineer neemt kennis, context en eerdere projectervaring mee. Een AI-agent heeft die informatie nodig om betrouwbaar werk te leveren.

Wat laat het onderzoek zien?

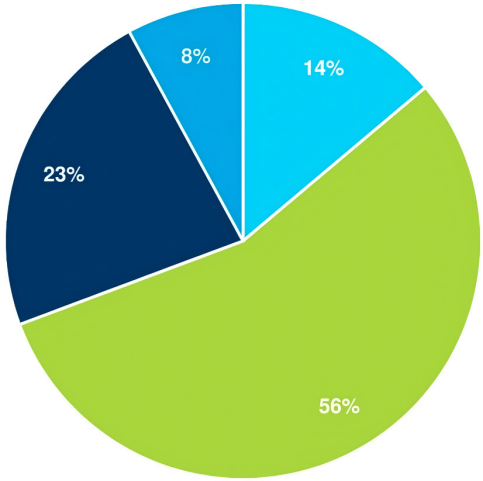
Veel organisaties zien dat de rol van engineers verandert, maar missen nog de kennis en vaardigheden om die verandering goed op te vangen. Daarnaast moeten organisaties opnieuw nadenken over opleiding, begeleiding en de rol van junior engineers.



Figuur 5

In hoeverre vormt een gebrek aan vaardigheden een belemmering voor het adopteren van agentic AI?

- Grote skills gap: het IT-team weet nog weinig van agentic AI en de inzet ervan.
- Medium skills gap: het IT-team experimenteert, maar gebruikt agentic AI nog niet optimaal.
- Kleine tot geen skills gap: het IT-team weet agentic AI goed in te zetten.
- Niet van toepassing: de organisatie is niet bezig met de adoptie van agentic AI.

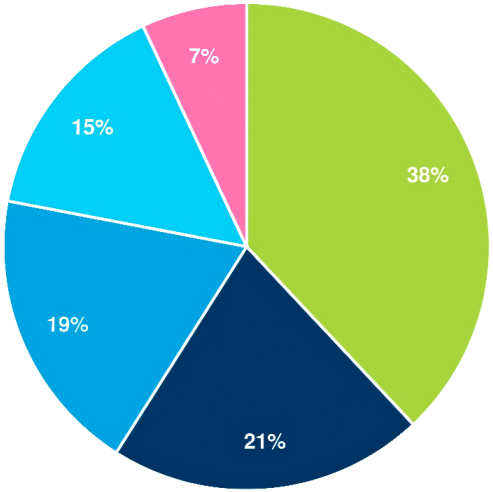


Door afronding tellen deze percentages niet exact op tot 100 procent.

Figuur 6

Hoe zie jij de rol van de software engineer veranderen door de komst van AI-agents?

- De focus verschuift naar beoordelen: de AI schrijft de code, terwijl de engineer valideert, test en bijstuurt (engineer als reviewer).
- De engineer werkt als regisseur van de agent: ontwerpt en beheert de AI-workflows zelf, bewaakt het gedrag van de agents en grijpt in waar de automatisering tekortschiet (engineer als agent-regisseur).
- De rol blijft in de kern gelijk, maar de output verandert: engineers schrijven minder zelf, maar blijven volledig eigenaar van de kwaliteit en werking (engineer als vakman).
- De focus verschuift naar het begin van het proces: het nauwkeurig definiëren van requirements, architectuur en kaders wordt het belangrijkste werk (engineer als specificatieauteur).
- Er ontstaat een tweedeling: een deel van de engineers specialiseert zich als AI-architect, terwijl het andere deel zich zal richten op uitvoerend werk met sterke AI-ondersteuning (gesplitst).



Hoewel zeven op de tien IT-beslissers een grote of middelgrote skills gap zien, noemt slechts **29 procent** een tekort aan expertise als een van de belangrijkste risico's bij de inzet van agentic AI voor softwareontwikkeling. Hoewel de vragen verschillende aspecten meten en de antwoorden niet rechtstreeks aan elkaar kunnen worden gekoppeld, wijst het verschil erop dat een kennisachterstand niet altijd als een van de belangrijkste adoptierisico's wordt herkend.

Vier aandachtsgebieden voor kennisontwikkeling

Info Support onderscheidt de volgende aandachtsgebieden voor kennisontwikkeling over agentic engineering:

- **AI-geletterdheid:**

begrijpen wat AI-modellen wel en niet kunnen, zodat teams agents niet overschatten en beter weten wanneer menselijke controle nodig blijft.

- **Agentic workflow design:**

taken, context, tools, feedback en escalatie zo ontwerpen dat agents herhaalbaar en veilig kunnen samenwerken met teams.

- **Engineering excellence:**

sterke basisprincipes voor specificatie, architectuur, testen, security en operations blijven toepassen, zodat snelheid van ontwikkeling niet ten koste gaat van kwaliteit.

- **Lerend vermogen:**

ervaringen meten, kennis delen en werkwijzen daarop aanpassen. Dat is belangrijk, want **36 procent** van de Nederlandse IT-beslissers zegt dat teams intern soms AI-successen delen, maar zelden wat misgaat.

Joop Snijder, Head of AI, adviseert:

> *"AI kan veel traditionele instaptaken versnellen of zelfs overnemen. Dat betekent niet dat junior engineers overbodig worden, maar wel dat leren anders moet worden ingericht. Als juniors vooral output genereren, zonder genoeg begrip op te bouwen van ontwerpkeuzes, foutpatronen en onderhoudbaarheid, ontstaat er een risico. Organisaties moeten daarom een nieuwe leercyclus creëren: werken met expliciete uitleg, pair engineering met mens en agent, gerichte reviews, simulaties en gecontroleerde verantwoordelijkheid. Training alleen is niet genoeg, het dagelijkse werk moet zelf ook een leeromgeving blijven."*

5. Op hoeveel gehuurde grond wil je bouwen?

Digitale soevereiniteit wordt een strategische ontwerpkeuze

Veel organisaties starten met publieke AI-diensten. Dat is logisch: teams kunnen er snel mee aan de slag en krijgen direct veel mogelijkheden. Maar zodra AI een vaste plek krijgt in bedrijfskritische softwareontwikkeling, groeit ook de afhankelijkheid van modellen, platformen en leveranciers.

Wat betekent dit voor organisaties?

Hoe autonomer AI wordt, hoe belangrijker regie wordt. Organisaties moeten daarom bewust bepalen waar ze zelf regie willen houden. Niet elke toepassing vraagt om maximale onafhankelijkheid, maar elke organisatie moet kunnen kiezen welk model ze gebruikt, waar dat model draait en welke data het mag verwerken. Digitale soevereiniteit betekent dus niet dat je alles zelf moet doen. Het gaat vooral om keuzevrijheid en wendbaarheid.

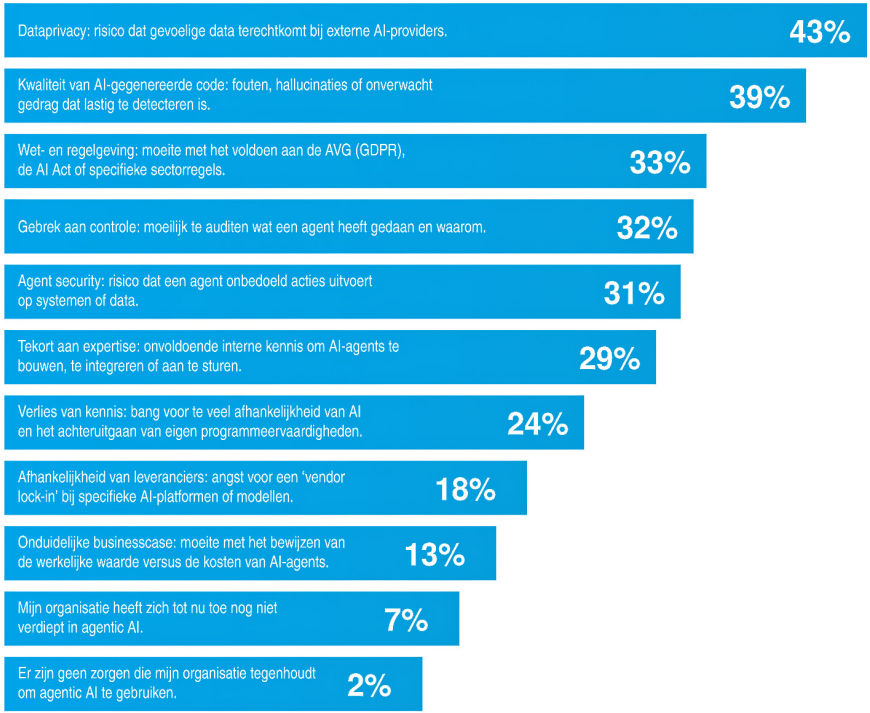
Wat laat het onderzoek zien?

Organisaties maken zich vooral zorgen over dataprivacy, kwaliteit en regelgeving. De angst voor vendor lock-in komt minder vaak naar voren, terwijl de strategische impact ervan groot kan zijn. Wie essentiële engineeringworkflows volledig rond één provider ontwerpt, bouwt een deel van zijn ontwikkelvermogen op gehuurde grond.

Figuur 7

Wat zijn de belangrijkste risico's of zorgen rond agentic AI bij softwareontwikkeling?

Respondenten konden meerdere antwoorden aanvinken.



Een pragmatisch soevereiniteitsmodel

Vijf niveaus van digitale soevereiniteit, van snelheid naar maximale controle.

NIVEAU	INRICHTING	GESCHIKT VOOR	BELANGRIJKSTE AFWEGING
1 **Publiek**	Direct gebruik van publiek AI-model of developertool	Laagrisicotaken en niet-vertrouwelijke context	Maximale snelheid, hoge externe afhankelijkheid
2 **Enterprise public**	Zakelijke AI-omgeving, contractuele waarborgen en centrale beheersmaatregelen	Breed intern gebruik met beheersbare data	Balans tussen toegankelijkheid en controle
3 **Hybride**	Meerdere modellen en routes per data- of risicoklasse	Organisaties met verschillende vertrouwelijkheidsniveaus	Portabiliteit en gerichte risicoreductie
4 **Private hosted**	Open of commercieel model in beheerde private omgeving	Vertrouwelijke engineeringcontext of sectorspecifieke eisen	Meer controle, hogere beheer- en integratielast
5 **Volledig soeverein**	Model, infrastructuur en operatie onder maximale eigen of Europese controle	Zeer kritieke processen en strategische autonomie	Maximale onafhankelijkheid, hoge investering en expertisebehoefte

MEER CONTROLE EN AUTONOMIE →

Frank Thiele
Unit Manager

> *"Een verstandige strategie begint met classificeren: welke data, codebases, applicaties en processen mogen via welke route worden verwerkt? Daarna kun je kritieke onderdelen ontkoppelen, modelkeuzes abstracter maken en alternatieven testen. Het doel is wendbaarheid. Kun je een model vervangen als kosten, kwaliteit, regelgeving of geopolitiek daarom vragen? Kunnen teams met vertrouwelijke informatie veilig met agentic engineering werken? En blijft de organisatie eigenaar van haar engineeringkennis, instructies, evaluaties en kwaliteitsdata? Daar zit de echte strategische keuze."*

6. Van pilot naar schaalbare engineeringpraktijk

Van koplopers naar vaste werkwijze

Veel organisaties geven hun ontwikkelteams vandaag toegang tot AI-tools en verwachten dat het gebruik daarna vanzelf groeit. In de praktijk blijft AI dan vaak hangen bij enthousiaste koplopers die er hun eigen productiviteit mee verhogen, terwijl een organisatiebrede werkwijze uitblijft.

Wat betekent dit voor organisaties?

AI opschalen gaat niet alleen over tooling, maar ook over de organisatie eromheen. Agentic engineering levert pas structureel waarde op wanneer teams dezelfde kwaliteitskaders gebruiken, kennis actief delen en AI onderdeel maken van architectuur, platformen en ontwikkelprocessen.

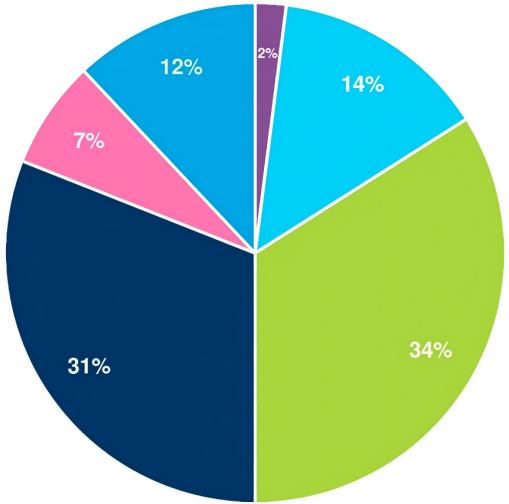
Wat laat het onderzoek zien?

Veel organisaties zitten in die overgangsfase. Richtlijnen zijn er vaak al, en koplopers tonen wat AI kan opleveren. Tegelijk blijven actieve governance, kennisdeling en een vaste plek in de dagelijkse engineeringpraktijk nog achter.

Figuur 8

In hoeverre zijn er duidelijke afspraken voor gebruik van en toezicht op AI-agents?

- Geen: AI-agents worden ingezet zonder formele regels of toezicht.
- Informeel: er zijn impliciete normen, maar niets is vastgelegd.
- Beleid aanwezig: er zijn richtlijnen, maar geen actief toezicht op naleving.
- Actieve controle: richtlijnen én toezicht via logging, audits en duidelijke rollen.
- Geoptimaliseerd: governance wordt continu bijgesteld op basis van incidenten en mogelijkheden.
- Er worden momenteel geen AI-agents ingezet.



Joop Snijder
Head of AI

“De valkuil is dat organisaties governance als los programma naast de deliverypraktijk zetten. Dan krijg je beleid op papier, maar verandert er weinig in het dagelijkse werk. Wij zien governance liever als onderdeel van goed engineeringontwerp: met duidelijke verantwoordelijkheden, uitvoerbare controles, meetbare kwaliteitscriteria en feedback uit echte projecten. Een kerngroep of enablementteam kan daarbij helpen, zolang die dicht op de teams werkt. Niet als centraal loket voor alle antwoorden, maar als versneller die patronen verzamelt, herbruikbare bouwblokken ontwikkelt en teams helpt om kwaliteit en adoptie zichtbaar te maken.”

Een praktische aanpak in vijf stappen

Een praktische aanpak voor het opschalen van agentic engineering bestaat uit vijf stappen:

1. Discover:

Breng samen in kaart waar de organisatie staat, welke use cases waardevol zijn en waar de grootste risico's zitten.

2. Prove:

Test de aanpak in de afgebakende pilots, met duidelijke kwaliteitscriteria en een eerste versie van het engineeringharnas.

3. Enable:

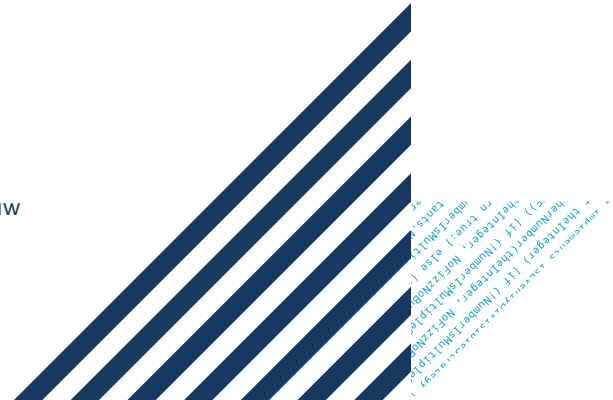
Help teams om zelfstandig met agentic engineering te werken via training, coaching, templates en kennisdeling.

4. Industrialize:

Maak de werkwijze onderdeel van de deliverypraktijk, met platformintegratie, CI/CD-controles, policies en observability.

5. Scale:

Hergebruik patronen over teams heen, stuur op metrics en bouw een organisatiebrede capability op.



Conclusie

Van individueel gebruik naar een volwassen praktijk

De eerste stap is gezet: AI is aanwezig in softwareontwikkeling. De echte opgave is nu om van individueel gebruik door te groeien naar een volwassen engineeringpraktijk. Dat vraagt om samenhang: een gedeeld feitenbeeld, sterk vakmanschap, duidelijke kwaliteitskaders, nieuwe vaardigheden, bewuste platformkeuzes en kennisdeling over teams heen. Dit beeld sluit aan bij ons bredere AI-adoptieonderzoek 2026, waarin dezelfde governance gap zichtbaar wordt: veel AI-activiteit op de werkvloer, maar weinig sturing van bovenaf.

Wie deze stappen goed zet, bouwt sneller én toekomstbestendiger. Organisaties die dit begrijpen moderniseren software consistenter, beheersen complexiteit beter en halen meer uit schaarse expertise. De vraag is niet of AI-agents onderdeel worden van softwareontwikkeling, maar of organisaties ze inbedden op een manier die betrouwbaar genoeg is voor de software waarop hun bedrijf draait.

Aan de slag met agentic engineering

Waar onze experts bij helpen

Reality check

Wil je de reality check niet alleen uitvoeren? Onze AI-experts helpen organisaties om samen scherp te krijgen waar ze staan en waar de eerste verbeterstappen liggen.

AI coding harness

Onze engineers helpen organisaties een AI coding harness in te richten binnen bestaande pipelines en platformen, met kaders die technisch worden afgedwongen.

Opschalen

Van discover en prove naar enable, industrialize en scale: samen maken we agentic engineering onderdeel van de dagelijkse engineeringpraktijk.

Over Info Support

Info Support biedt end-to-end IT-oplossingen die jouw organisatie versterken. Onze ruim 500 experts nemen verantwoordelijkheid voor het volledige traject. Jouw kritische IT in veilige handen, gebouwd op vakmanschap, versterkt met AI in de hele lifecycle.

Wij helpen organisaties om businessvraagstukken te vertalen naar AI-functionaliteit die geïntegreerd wordt in bestaande applicaties, dataplatformen en bedrijfsprocessen. Met een beproefde aanpak en hoogwaardige trainingen zorgen we ervoor dat jouw organisatie klaar is voor de toekomst en optimaal kan excelleren en innoveren.

- **AI**
- **Cloud**
- **Data**
- **Software engineering**
- **Training**

Hoofdkantoor Nederland
Kruisboog 42
3905 TG Veenendaal

T +31 (0)318 552 020

Hoofdkantoor België
Generaal de Wittelaan 17 | bus 30
2800 Mechelen

T +32 (0)15 28 63 70
www.infosupport.com

Solid Innovator