

WHITEPAPER

AI in softwareontwikkeling: kansen, risico's en onze aanpak

Hoe wij AI inzetten als versneller — zonder concessies te doen aan kwaliteit, controle en klantwaarde

Juni 2026

INLEIDING

De wereld verandert. Wij ook.

Kunstmatige intelligentie is geen toekomstige technologie meer — het is nu. Dagelijks zien wij hoe AI-tools de manier waarop software wordt ontwikkeld, getest en onderhouden fundamenteel verschuiven. De vraag is niet óf je als softwarebedrijf met AI aan de slag gaat, maar hoe je dat verantwoord doet.

In dit whitepaper zetten wij onze visie uiteen. Wat zien wij als de kansen van AI voor softwareontwikkeling? Wat zijn de risico's — en waarom nemen wij die serieus? En hoe vertaalt zich dat concreet naar de keuzes die wij maken: in onze technologie, onze werkwijze, en onze relatie met klanten.

AI is voor ons een versneller, geen vervanger. Snelheid heeft alleen waarde als de kwaliteit en controle geborgd blijven.

ONZE VISIE

AI als onderdeel van ons vakmanschap

Wij ontwikkelen standaardoplossingen in TypeScript en Kotlin, die wij snel kunnen uitbreiden voor een grote en groeiende groep klanten. Dat vraagt om een consistent ontwikkelproces: reproduceerbaar, controleerbaar en kwalitatief hoogwaardig. AI past in dat beeld — mits je het bewust inzet.

Onze visie rust op drie pijlers:

- AI versnelt, mensen beslissen. Wij gebruiken AI-tools zoals Cursor en GitHub Copilot om sneller code te schrijven, patronen te herkennen en repetitief werk te automatiseren. Maar architectuurkeuzes, kwaliteitsbeoordelingen en klantbeslissingen blijven mensenwerk.
- Standaardisatie is onze kracht. Juist omdat wij standaardoplossingen bouwen, kunnen wij AI effectief voeden met onze eigen codebase, conventies en patronen. Een AI die onze architectuur 'kent', presteert significant beter dan een generieke assistent.
- Controle is niet optioneel. Meer features bouwen met AI betekent ook meer code die sneller kan groeien dan de beheerbaarheid toelaat. Daarom investeren wij parallel in code reviews, testdekking en CI/CD-guardrails.

K A N S E N

Wat AI concreet mogelijk maakt

Sneller nieuwe features leveren

Het grootste directe voordeel van AI in ons ontwikkelproces is tempo. De literatuur spreekt over 30 tot 50 procent tijdwinst bij taken als componenten genereren, boilerplate uitschrijven en unit tests aanmaken. Die percentages zien wij in de praktijk nog niet terug — de werkelijke winst is op dit moment nog niet goed te bepalen. Wat wij wél zien: de productiviteit van de programmeur stijgt significant. Hoeveel precies, valt te bezien. AI geeft veel ideeën en kan debuggen. Dat levert tijdwinst op, en daardoor kan de developer werkzaamheden verschuiven naar werk waar zijn of haar inzicht het verschil maakt. Hoeveel dit onder de streep oplevert, gaan we de komende maanden zien — en meten.

Concreet zien wij winst op:

- Frontend-componenten: van ontwerp naar werkende TypeScript-code in een fractie van de tijd.
- Backend-integraties: Kotlin-services en API-koppelingen die met AI-assistentie sneller worden opgezet en gedocumenteerd.
- Testsuites: AI kan op basis van bestaande code automatisch testscenario's voorstellen, wat de dekking verhoogt zonder evenredige tijdsinvestering.

Betere ondersteuning van klanten

AI maakt het ook mogelijk om klantondersteuning te verbeteren zonder de organisatie navenant te laten groeien. Denk aan AI-gestuurde analyse van supportverzoeken, automatische classificatie van meldingen op ernst en prioriteit, en slimmere rapportages over systeemgedrag. Daardoor kunnen onze medewerkers zich richten op de complexe vragen waar menselijk inzicht écht het verschil maakt.

Proactief onderhoud

Traditioneel onderhoud is reactief: een klant meldt een probleem, en dan wordt het opgelost. AI verschuift dit model richting preventie. Door loganalyse, patroonherkenning en voorspellende monitoring kunnen wij problemen signaleren vóórdat ze uitval veroorzaken. Dit verhoogt de betrouwbaarheid van onze oplossingen structureel.

RISICO'S

De keerzijde: risico's die wij serieus nemen

De hype rond AI heeft een schaduwzijde. Organisaties die klakkeloos AI-tools omarmen zonder bewuste keuzes te maken, lopen reële risico's. Wij benoemen ze hier expliciet — niet om af te schrikken, maar omdat we geloven dat eerlijkheid de basis is van goed vakmanschap.

Risico 1 — Kwaliteitsverlies door onbeheerste codegroel

- AI genereert snel veel code. Zonder strakke review-discipline en testcoverage ontstaat er 'AI-slop': code die syntactisch klopt maar architecturaal niet past.
- In een standaardplatform dat door tientallen klanten wordt gebruikt, kan één slecht doordachte aanpassing verstreckende gevolgen hebben.
- Onze aanpak: verplichte code reviews ook voor AI-gegenereerde code, en harde coverage-drempels in onze GitLab CI/CD-pipelines.

Risico 2 — Verlies van architectureel begrip

- Naarmate meer code door AI wordt geschreven, neemt het risico toe dat ontwikkelaars de diepere logica minder goed begrijpen. Dat is gevaarlijk bij onderhoud en uitbreiding.
- Wij zien dit als een cultuurvraagstuk: AI is een hulpmiddel voor de ontwikkelaar, niet een vervanger van diens kennis.
- Onze aanpak: architectuurprincipes expliciet op papier, en AI-gebruik wordt begeleid door senior engineers.

Risico 3 — Afhankelijkheid van externe AI-diensten

- De AI-tools die wij gebruiken zijn externe diensten. Prijzen, beschikbaarheid en mogelijkheden kunnen veranderen — soms snel.
- Een workflow die volledig afhankelijk is van één AI-provider is kwetsbaar.
- Onze aanpak: toolkeuzes worden gemaakt op basis van integratiemogelijkheden, niet op hype. Wij houden onze werkwijze toolonafhankelijk waar mogelijk.

Risico 4 — Beveiliging en privacy van klantdata

- AI-codeerassistenten die context meesturen naar externe API's vormen een potentieel privacy-risico, zeker als klantdata of bedrijfsgevoelige logica in de context zit.
- Onze aanpak: duidelijk beleid over welke context wij wel en niet delen met AI-diensten, en waar nodig gebruik van on-premise of privacy-first alternatieven.

ONZE AANPAK

Hoe wij AI verantwoord inzetten

Standaardoplossingen als fundament

Juist omdat wij standaardoplossingen bouwen voor een brede klantgroep, zijn wij in een goede positie om AI goed in te zetten. Onze codebase is consistent, onze patronen zijn vastgelegd, en onze architectuurprincipes zijn expliciet. Dat maakt het mogelijk om AI te voeden met relevante context — en daarmee de kwaliteit van de output structureel te verhogen.

Een AI die werkt met een doordachte TypeScript-componentenbibliotheek en bekende Kotlin-servicepatronen, maakt andere fouten dan een AI die in het wilde weg gegenereerde code produceert. Standaardisatie is onze bescherming.

Tempo én controle

Wij verhogen het ontwikkeltempo door AI in te zetten voor het schrijven van code, het genereren van tests en het versnellen van code reviews. Maar wij doen dit binnen een framework van bewuste controles:

- GitLab CI/CD met automatische kwaliteitschecks bij elke commit.
- Linear voor gestructureerde feature-planning, zodat AI-gegenereerde features altijd traceerbaar zijn naar een klantbehoefte.
- Peer reviews als standaard, ook voor AI-ondersteunde code.
- Testdekking als harde drempel, niet als nice-to-have.

AI inzetten op bouwen én op onderhouden

Er is een onderscheid dat vaak over het hoofd wordt gezien: AI ingezet op het bouwen van features versnelt ook de aangroei van onderhoudslast. Elke feature die je sneller shipt, vraagt jaarlijks onderhoud — ongeacht hoe snel hij gebouwd is. Een team dat AI puur gebruikt om sneller te bouwen, bereikt het capaciteitsplafond van wat het kan onderhouden gewoon eerder.

De vuistregel is helder: voor elke week bouwen reserveer je ruwweg één dag per jaar aan onderhoud. AI verlaagt de bouwkosten, maar verlaagt die onderhoudsdag niet automatisch. Meer features shippen in dezelfde tijd betekent: de onderhoudslijn stijgt steiler.

AI gericht op bouwen brengt je sneller bij het plafond. AI gericht op onderhoud verhoogt dat plafond.

Daarom richten wij AI bewust ook op de onderhoudskant: het bijhouden van dependency-updates, het uitvoeren van migraties als een externe dienst wijzigt, het herstructureren van code, en het veilig en goedkoop maken van het verwijderen van features die hun waarde hebben gehad. Hetzelfde gereedschap, tegenovergesteld resultaat — afhankelijk van waar je het op richt.

Voor ons als bouwer van standaardoplossingen is dit extra relevant. Een groeiende klantgroep betekent dat de onderhoudslast per definitie toeneemt. Wij bewaken actief de verhouding tussen nieuw werk en onderhoud, en gebruiken AI aan beide kanten van die balans.

De rol van onze frontend developer

AI-tools kunnen steeds meer frontend-code genereren. Dat verandert de rol van een frontend developer — maar schrapt die niet. De meerwaarde van een ervaren frontend developer zit in het beoordelen van architectuurkeuzes, het bewaken van de gebruikerservaring, en het herkennen van AI-gegenereerde code die syntactisch klopt maar semantisch tekortschiet.

Wij geloven dat een sterke frontend developer mét AI aanzienlijk productiever is dan zonder — hoeveel precies, gaan onze eigen metingen de komende maanden uitwijzen. De vraag is niet: 'Hebben we dit nog nodig?' — maar: 'Gebruikt deze persoon AI als multiplier?'

De waarde van een goede developer zit niet in het typen van code. Die zit in het begrijpen van het probleem en het bewaken van de oplossing.

VOOR ONZE KLANTEN

Wat betekent dit voor u?

Alles hierboven gaat over hoe wij werken. Maar de vraag die er voor u toe doet, is een andere: wat merkt u ervan? In gewone taal komt het hierop neer:

- Wensen worden sneller werkelijkheid. Doordat onze ontwikkelaars routinewerk aan AI kunnen overlaten, komt er tijd vrij voor wat u vraagt: nieuwe functionaliteit, verbeteringen en maatwerkwenen worden sneller opgepakt.
- De kwaliteit blijft geborgd door mensen. Elke aanpassing — ook als AI heeft meegeschreven — wordt beoordeeld door een ervaren ontwikkelaar voordat die u bereikt. U krijgt nooit ongecontroleerde, machinaal gegenereerde software.
- Minder verstoringen, snellere antwoorden. AI helpt ons problemen te zien vóórdat u er last van heeft, en supportvragen sneller bij de juiste persoon te krijgen. De complexe vragen blijven mensenwerk — juist daar maakt persoonlijk contact het verschil.
- Uw gegevens blijven beschermd. Wij hebben duidelijk beleid over welke informatie wel en niet met AI-diensten wordt gedeeld. Klantdata en bedrijfsgevoelige informatie vallen daar niet onder.

AI in onze oplossing zelf

De vraag die wij steeds vaker krijgen: komt er ook AI in de oplossing die u zelf gebruikt? Het eerlijke antwoord: dat zijn we nu aan het verkennen — voorzichtig, en tegelijk met een open blik. Twee sporen zijn al concreet. We zijn ver met een roosterautomaat voor capaciteitsplanning, die planners helpt om sneller tot een kloppend rooster te komen. En we onderzoeken hoe AI ons kan ondersteunen in de support, zodat uw vragen sneller en beter beantwoord worden.

Tegelijk zijn we reëel over de grenzen. Wat wij nog niet zien, is dat AI de volledige context kan overzien waarin uw planning tot stand komt: de afspraken, de uitzonderingen, de mensen en de organisatie erachter. Juist daar blijven wij onze eigen inzet op plegen. AI rekent en stelt voor; het begrijpen van uw situatie en het maken van de juiste afweging blijft het werk van onze mensen — en van uw planners.

CONCLUSIE

Versnellen met verstand

AI verandert softwareontwikkeling fundamenteel — en dat is goed nieuws voor bedrijven die er bewust mee omgaan. Wij omarmen AI als versneller van ons ontwikkelproces, als ondersteuner van onze klantdienstverlening, en als instrument voor slimmer onderhoud.

Maar wij doen dit met open ogen. De risico's van onbeheerste codegroel, verlies van begrip, externe afhankelijkheden en privacyvraagstukken zijn reëel. Wij adresseren ze proactief, niet nadat ze problemen hebben veroorzaakt.

Onze standaardoplossingen zijn daarbij ons grootste voordeel: een consistente codebase maakt AI effectiever, en een gestructureerde werkwijze maakt de output controleerbaar.

Wij bouwen software die klanten kunnen vertrouwen — en AI helpt ons dat beter en sneller te doen. Zonder concessies aan kwaliteit.

BIJLAGE

Voor wie meer wil weten: de techniek

Deze bijlage is bedoeld voor lezers die willen weten h oe wij het hierboven beschreven beleid technisch waarmaken — bijvoorbeeld functioneel beheerders of IT-collega's die meekijken bij leveranciersbeoordeling. Voor het begrijpen van onze visie is deze bijlage niet nodig.

- Technologie. Onze standaardoplossingen zijn gebouwd in TypeScript (frontend) en Kotlin (backend). Die consistente basis maakt AI-assistentie effectiever: de AI werkt binnen vastgelegde patronen en een doordachte componentenbibliotheek, in plaats van in het wilde weg code te genereren.
- AI-gereedschap. Wij werken met AI-codeerassistenten zoals Cursor en GitHub Copilot, gevoed met onze eigen codebase, conventies en architectuurprincipes. Toolkeuzes maken wij op integratiemogelijkheden, niet op hype; onze werkwijze houden wij waar mogelijk toolonafhankelijk om niet afhankelijk te worden van   n leverancier.
- Kwaliteitsborging. Elke wijziging doorloopt onze GitLab CI/CD-pipeline met automatische kwaliteitschecks en harde drempels voor testdekking. Peer review door een ervaren ontwikkelaar is verplicht — ook, en juist, voor code waar AI aan heeft meegeschreven.
- Traceerbaarheid. Featureplanning verloopt via Linear, zodat elke (AI-ondersteunde) aanpassing herleidbaar is naar een concrete klantbehoefte. Zo bouwen we nooit 'omdat het kan', maar omdat het iets oplost.
- Onderhoud. Wij zetten AI ook in aan de onderhoudskant: dependency-updates, migraties bij wijzigingen in externe diensten, herstructurering van code en het veilig uitfasen van functionaliteit. Vuistregel die wij hanteren: elke week bouwen vraagt ruwweg   n dag onderhoud per jaar — AI moet dus aan b  ide kanten van die balans werken.
- Privacy en datadeling. Wij hanteren expliciet beleid over welke context met externe AI-diensten gedeeld mag worden. Klantdata en bedrijfsgevoelige logica vallen daarbuiten; waar nodig kiezen wij voor on-premise of privacy-first alternatieven.

Vragen of reacties naar aanleiding van dit whitepaper?

Neem contact op via uw vaste contactpersoon of stuur een e-mail naar info@uwbedrijf.nl