Advies over GitHub Copilot en Claude Code

GitHub Copilot is een in de IDE (ontwikkelomgeving) geïntegreerde assistent die fungeert als intelligente 'autocomplete' voor snelle, incrementele code-suggesties tijdens het typen. Claude Code is daarentegen een 'agentic' (autonome) tool die vanuit de terminal werkt, de context van een volledige repository kan lezen en zelfstandig stappenplannen maakt voor complexe wijzigingen over meerdere bestanden.

Wanneer we de inzet van deze specifieke tools afzetten tegen de kaders voor AI-compliance binnen onder organisatie (zoals de AI Act, AVG en het huidige AI-beleid), komen er vier risicogebieden naar voren:

1. Schaduw-IT en Datagovernance (Privacy & AVG)

Onze praktijk: Ontwikkelaars kunnen in de werkomgeving vrijelijk gebruik maken van ongeautoriseerde AI-tools ("Shadow AI") om hun productiviteit te verhogen. Hierbij plakken ze mogelijk (bron)code, API-sleutels of klantgegevens in de prompts van de AI. Uit onderzoek blijkt dat repositories die Copilot gebruiken 40% vaker uitgelekte 'secrets' (zoals wachtwoorden) bevatten dan traditionele projecten.

Uit recent onderzoek blijkt dat shadow AI niet alleen wijdverspreid is, maar ook aantoonbaar risicoverhogend werkt: repositories waarin AIcoding assistants worden gebruikt laten 40% meer datalekken zien en in 6,4% van de gevallen duiken daadwerkelijk gevoelige gegevens op. Bovendien groeit het gebruik van ongeautoriseerde AI-tools met 120% per jaar, wat laat zien dat dit geen randverschijnsel is maar een structurele ontwikkeling. -tools met 12-0% per jaar, wat laat zien dat dit geen randverschijnsel is maar een structurele ontwikkeling.

De AI Act en de AVG vereisen bovendien strikte datagovernance en "privacy by design". Het gebruik van gratis versies van GitHub Copilot of Claude Code waarbij code in de trainingsdata van de leverancier belandt, is een direct risico. Vanuit de rol van AI compiance officer raden we dan ook aan om uitsluitend beveiligde enterprise-accounts af te sluiten (waarbij data niet voor training wordt gebruikt) en het ongecontroleerde gebruik van dit soort AI-assistenten te verbieden.

2. Regie en gelijke toegang

Binnen de organisatie werken ook collega's op inhuurbasis. Een deel van hen beschikt via hun eigenlijke werkgever over enterprise-accounts voor AI-assistenten, die in de praktijk kunnen worden ingezet bij werkzaamheden voor de provincie. Los van de formele vraag of dit is toegestaan, is het begrijpelijk dat externen gebruikmaken van de voor hen beschikbaar gestelde middelen om hun werk efficiënt uit te voeren.

Deze situatie kan echter leiden tot scheefgroei. Wanneer het gebruik van AI-assistenten resulteert in hogere productiviteit, ontstaan verschillen in tijdsbesteding en werkdruk tussen medewerkers met en zonder toegang tot deze tools. Dit kan het gevoel geven van ongelijke of oneerlijke werkomstandigheden, met name wanneer interne medewerkers bewust worden beperkt.

Daarnaast raakt het gebruik van externe enterprise-accounts aan regie en toezicht. Deze vallen buiten de directe governance van provincie Gelderland, terwijl de gegenereerde output wel onderdeel wordt van de systemen en processen. Dit benadrukt de noodzaak van heldere kaders en een organisatiebrede aanpak.

3. Auteursrecht en Aansprakelijkheid (Intellectueel Eigendom)

Modellen voor code-generatie zijn getraind op miljarden regels open-source code (voorzien van licenties zoals MIT of GPL). De modellen strippen echter de vereiste bronvermeldingen uit de gegenereerde code. In de VS is inmiddels bepaald dat pure AI-gegenereerde code geen auteursrechtelijke bescherming geniet omdat de menselijke auteur ontbreekt. Provincie Gelderland draagt dus de volledige juridische aansprakelijkheid als onze AI-gegenereerde code inbreuk maakt op andermans auteursrecht, maar krijgt zelf geen IE-bescherming voor die code. De AI Act benadrukt dat het de verantwoordelijkheid van de

organisatie is om de herkomst van data (data lineage) en licenties te documenteren. Organisaties moeten daarom tools selecteren waarvan de leverancier vrijwaring (indemnity) biedt voor IE-inbreuken, hoewel deze clausules vaak beperkt zijn.

4. Outputkwaliteit, Security en Automation Bias

AI-assistenten genereren nog altijd met grote regelmaat onveilige code of kwetsbare patronen (zoals SQL-injecties), mede doordat hun trainingsdata verouderde of onveilige voorbeelden bevat. Daarnaast kunnen tools als Claude Code en GitHub Copilot 'hallucineren' door niet-bestaande softwarepakketten voor te stellen, wat reële supply chain-risico's introduceert. Juist omdat de gegenereerde code er overtuigend en ogenschijnlijk foutloos uitziet, ligt automation bias op de loer en bestaat het risico dat ontwikkelaars deze zonder voldoende controle accepteren.

In het licht van onder andere de AI Act, die expliciet technische robuustheid, veiligheid en aantoonbaar menselijk toezicht vereist, is het daarom urgent om een gecontroleerde pilot te starten met zowel GitHub Copilot als Claude Code. Deze pilot dient plaats te vinden binnen beveiligde enterprise-accounts en met een geselecteerde groep ontwikkelaars die systematisch toetst welke tool voldoet aan onze kwaliteits- en securitystandaarden. Tegelijkertijd biedt dit traject de noodzakelijke basis om heldere werkafspraken vast te leggen over de '*human in the loop*', AI-geletterdheid te versterken en te bepalen hoe deze tools verantwoord en compliant in onze ontwikkelprocessen worden ingezet, inclusief de opschaling van kwaliteitsborging om de toegenomen stroom AI-gegenereerde code effectief te controleren.

5. Transparantie en Traceerbaarheid

Als ontwikkelaars AI-tools gebruiken om sneller code te schrijven, wordt het lastiger om te zien wie wat heeft gemaakt: de mens of de AI. Dat lijkt onschuldig, maar wordt een probleem als je later moet controleren of alles volgens de regels is gedaan (bijvoorbeeld voor privacy- of security-eisen zoals SOC2, HIPAA of de AVG).

Die regels verwachten namelijk dat je kunt uitleggen en aantonen wat er is gebeurd: Wie heeft de code geschreven? Waar komt bepaalde logica vandaan? Zijn er veilige keuzes gemaakt?

Met AI vervaagt dat onderscheid, en zonder extra maatregelen kun je dat achteraf moeilijk reconstrueren. Daarom is traceerbaarheid (logging) belangrijk: je moet bijhouden welke vragen (prompts) aan de AI zijn gesteld, welke antwoorden/code de AI gaf en hoe die vervolgens is gebruikt.

Als de pilot succesvol blijkt, ontstaat de vervolgvraag hoe we het gebruik van AI-assistenten structureel en beheerst gaan inrichten binnen onze organisatie. Dit raakt met name aan vraagstukken rond governance en platforminrichting: hoe organiseren we een centrale controlelaag (bijvoorbeeld in de vorm van een AI-gateway) waarmee we het gebruik van deze tools kunnen sturen, beveiligen en monitoren bij bredere ingebruikname.