

JDriven Tech Radar

Najaar 2025

Onze kijk op recente ontwikkelingen in het veld

12e editie

**Commit to know.
Develop to grow.
Share to show.**





Commit. Develop. Share.

Voorwoord

Met veel trots presenteren we de 12e editie van de JDriven Tech Radar.

Zoals menig een zal hebben opgemerkt op congressen en social media zijn er twee onderwerpen die de gemoederen het afgelopen jaar hebben beziggehouden in ons vakgebied: AI en digitale soevereiniteit in Europa. In zo'n situatie is het verleidelijk om mee te gaan in het heersende sentiment en kritiek klakkeloos over te nemen of te verwerpen. De opzet van de totstandkoming van de JDriven Tech Radar beschermt ons hiervoor. Allereerst moeten we voor zowel een onderwerp als de mening over dat onderwerp naar elkaar toe onderbouwen of het op de radar moet komen. Is dat eenmaal besloten, dan is er een stijlguide die we gebruiken voor het uitschrijven van de blip-tekst, waarbij we eerst objectief beschrijven wat iets is en daarna subjectief onze mening erover geven. Dit onderscheid helpt meerdere perspectieven in acht te nemen en feiten en meningen uit elkaar te houden. Ten slotte worden aangedragen teksten behandeld zoals een merge request: collega's reviewen elkaars bijdragen. Op deze manier kunnen we trends volgen en genuanceerd en goed onderbouwd duiden. Volgens mij is dat ook deze keer gelukt.

Bedankt aan eenieder die deze editie mogelijk heeft gemaakt!



Jasper Bogers
Consultant JDriven



Introductie

Onze ervaren specialisten werken dagelijks mee aan tal van softwareprojecten in Nederland en zijn betrokken in wereldwijde community's. Halfjaarlijks komen wij vanuit JDriven bij elkaar om te bespreken wat wij aan nieuwe trends en ontwikkelingen zien. Deze trends proberen wij te vangen in een technologieradar. Elke editie van de radar laat verschuivingen zien t.o.v. een vorige editie. Een verschuiving kan betekenen dat wij een technologie interessanter zien worden waar van toepassing, of juist minder geschikt ongeacht de toepassing. Indien een trend niet meer voorkomt in een latere editie, dan zijn er geen nieuwe ontwikkelingen en/of ervaringen die ons eerdere beeld zouden hebben bijgesteld. In dit document willen we toelichten welke verschuivingen we de afgelopen periode hebben waargenomen, om op basis daarvan weer richting te geven aan wat wij inzetten en aanraden.

Tech Radar

Het idee voor het opstellen van een Tech Radar komt voort vanuit Thoughtworks. Zij dragen al langer periodiek met een radar hun visie uit op nieuwe trends en ontwikkelingen. Bovendien raden zij iedereen aan [een eigen radar op te stellen](https://www.thoughtworks.com/radar/byor) [https://www.thoughtworks.com/radar/byor].

Bij JDriven onderschrijven we dat. Het opstellen van een Radar is een leerzame en waardevolle ervaring waarin onderling kennis wordt gedeeld en een technisch bewustzijn wordt gecreëerd. Wij geloven erin dat specialisten zelf in staat moeten zijn om het gereedschap voor hun werkzaamheden samen te stellen. Met het opstellen van een radar faciliteer je discussies over technologie, om als organisatie de juiste balans te vinden in wat voor risico's en voordelen innovatie kan opleveren. Wij kunnen je helpen dit op te starten binnen je organisatie. Laat je teams elkaar inspireren tot innovatie en gezamenlijk komen tot een set aan technologieën en technieken die de ontwikkeling in je bedrijf versnellen.

Indeling

Een radar bestaat uit kwadranten en ringen, met daarin blips om interessante technologieën en technieken aan te duiden.

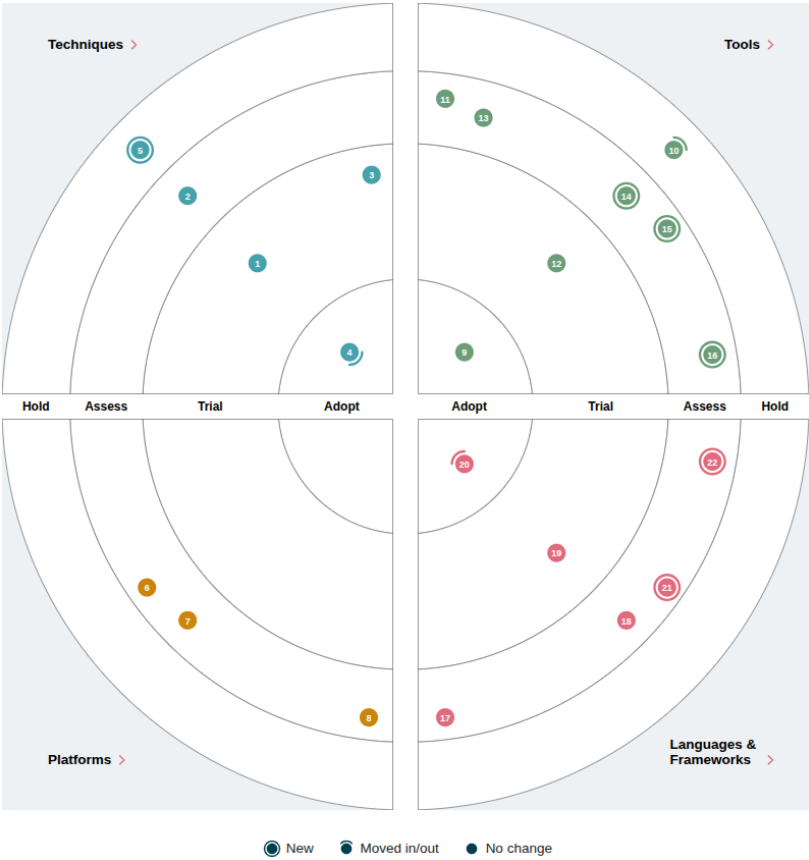
De kwadranten verdelen de verschillende onderwerpen in categorieën.

- **Talen & frameworks** die je ondersteunen bij de ontwikkeling van software
- **Platformen** waarop je software kunt uitvoeren
- **Technieken** die je helpen betere software te maken
- **Tools** ter ondersteuning van je ontwikkel- en delivery-proces

De ringen in elk kwadrant geven aan in welk stadium van adoptie wij denken dat die technologie zich bevindt.

- **Adopt** → Wij raden sterk aan deze technologie te gebruiken, waar van toepassing.
- **Trial** → Interessant om alvast ervaring mee op te doen (in een project dat het risico kan dragen).
- **Assess** → Goed om beter te begrijpen en toekomstige impact in te schatten, maar nog niet om toe te passen.
- **Hold** → Niet (meer) gebruiken.

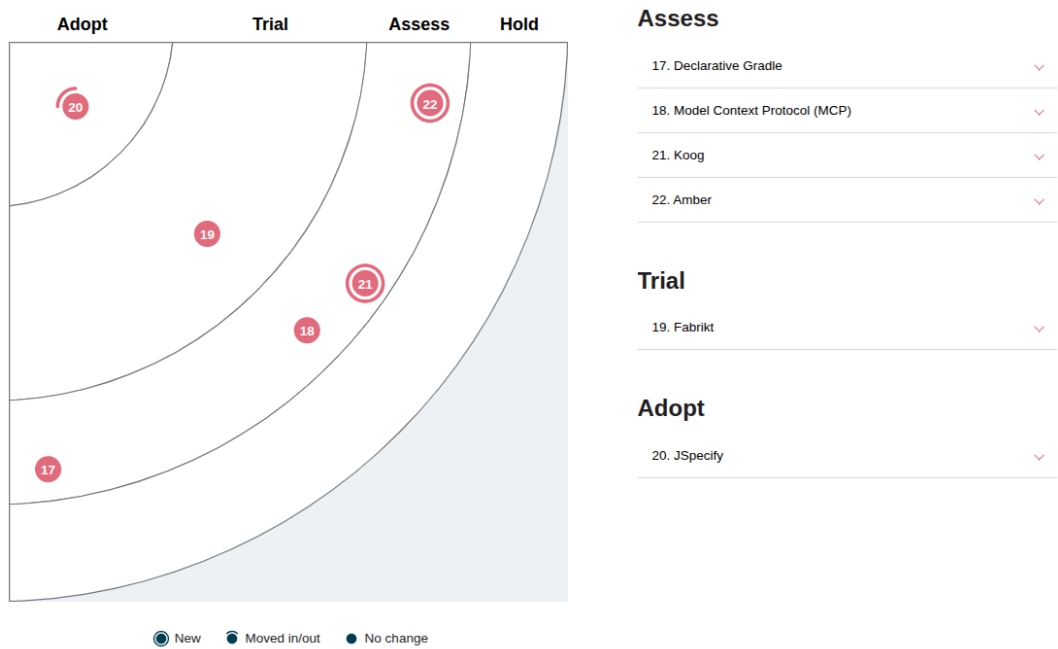
In de volgende secties zullen we onze kijk op de recente ontwikkelingen per kwadrant toelichten.



Tools	Talen & frameworks
ADOPT IntelliJ HTTP client	ADOPT JSpecify
TRIAL Nushell	TRIAL Fabrikt
ASSESS Anubis BlueDolphin Cursor IDE DeepSeek Maven 4	ASSESS Amber Declarative Gradle Koog Model Context Protocol (MCP)
HOLD JetBrains AI	HOLD -
Platforms	Technieken
ADOPT -	ADOPT Event Modeling
TRIAL -	TRIAL Embrace Shadow IT Passkeys
ASSESS AWS European Sovereign Cloud EU Cloud alternatives Proxmox	ASSESS AI Agents
HOLD -	HOLD Vibe coding



Talen & frameworks



JSpecify

ADOPT

JSpecify [<https://jspecify.dev/>] is een annotatie-framework dat de afhandeling van nullability in Java verbeterd door expliciete type-informatie te bieden, geïnspireerd door het nullability-systeem van Kotlin. Het helpt `NullPointerException`'s te voorkomen door nullability-contracten op compile-tijd te standaardiseren. Het project is al meer dan vijf jaar in ontwikkeling en is ontworpen door grote partijen zoals Google, JetBrains, Meta, Microsoft, Oracle en Uber. Met de release van versie 1.0 is de specificatie gestabiliseerd, wat een solide basis biedt voor adoptie. IntelliJ IDEA, Project Lombok, NullAway en het Checker Framework bieden momenteel al ondersteuning voor JSpecify.

Migration guides zijn beschikbaar om over te stappen van JSR-305- of Checker Framework-annotaties naar de JSpecify-varianten. Nu Spring Framework 7 JSpecify omarmt en bredere ondersteuning in tooling aanwezig is, is het de standaardkeuze geworden voor het verbeteren van null safety in Java. JSpecify heeft zich bewezen als een stabiele, productieklare oplossing met uitstekende tooling-ondersteuning en brede acceptatie in het Java-ecosysteem. We bevelen sterk aan om JSpecify te gebruiken voor nieuwe projecten en te overwegen voor bestaande codebases. Daarom plaatsen we het in de Adopt-ring.

Fabrikt

TRIAL

OpenAPI, voorheen bekend als Swagger, is een veelgebruikte interfacespecificatie voor het definiëren en documenteren van API's. De meest bekende tool die gebruikt wordt om op basis van een OpenAPI-specificatie code te genereren is waarschijnlijk OpenAPITools' openapi-generator. Fabrikt is een alternatief dat zich puur richt op het genereren van Kotlin-code op basis van de OpenAPI v3-specificatie. Het biedt ondersteuning voor zowel Spring, Micronaut als Ktor, en biedt bovendien een Gradle plugin.

In tegenstelling tot openapi-generator kan Fabrikt goed overweg met complexere structuren die OpenAPI v3 kent, met *anyOf* als bekendste voorbeeld. Fabrikt weet bovendien, mede dankzij de focus op OpenAPI v3 en Kotlin, openstaande issues en merge requests goed bij te benen; iets waar openapi-generator mee worstelt. Dit zijn redenen waarom wij aanraden om Fabrikt te gebruiken voor op OpenAPI v3 gebaseerde Kotlin code. Omdat Fabrikt complexere specificaties aankan en andere code kan genereren dan openapi-generator, raden we aan klein te beginnen en te evalueren wat de impact is van een migratie.

Amber

ASSESS

Amber [<https://amber-lang.com/>] is een moderne programmeertaal die compileert naar Bash shell-scripts. Het pakt de problemen aan van foutgevoelige en lastig te onderhouden shell-scripts door compile-time type checking, een moderne syntaxis vergelijkbaar met ECMAScript en de toevoeging van een uitgebreide standaardbibliotheek. Dit maakt shell-scripts betrouwbaarder en helpt runtime-fouten te voorkomen. Het project won in 2024 de prijs voor "The Best Engineering Project in Technology" aan de Universiteit van Wrocław. De taal ondersteunt vijf datatypes (Text, Num, Bool, Null en `Array`s), biedt moderne programmeerconstructen zoals functies, loops en condities, en integreert naadloos met bestaande Bash-scripts.

Amber bevindt zich momenteel in versie 0.4.0-alpha en is nog volop in ontwikkeling. Hoewel het concept veelbelovend is voor het verbeteren van shell-scripting, zijn er nog beperkingen, zoals het ontbreken van ondersteuning voor geneste arrays vanwege de beperkingen van Bash zelf, en het risico op incompatibele wijzigingen tussen versies. De taal is interessant om te volgen voor teams die veel gebruikmaken van shell-scripts, maar is nog niet rijp voor productie. Daarom plaatsen we Amber in de Assess-ring.

Declarative Gradle

ASSESS

Bij gebruik van Gradle zijn er veel verschillende manieren om je build-scripts in te richten. Dit kan voor veel verwarring zorgen, vooral als je met meerdere projecten werkt die verschillende manieren van werken hebben. Bij Gradle zijn ze zich hiervan ook bewust en daarom hebben ze eind 2023 aangekondigd begonnen te zijn met een experiment om Gradle declaratief te maken. Dit project heet **Declarative Gradle** [<https://declarative.gradle.org>] en is nog in een experimentele fase. Het doel van dit project is om een vaste structuur aan te bieden voor Gradle build-scripts, zodat deze makkelijker te lezen en te onderhouden zijn. Dit moet ook de herkenbaarheid tussen projecten verbeteren.

We plaatsen deze nieuwe standaard in de Assess-ring, omdat we het de moeite waard vinden ontwikkelingen op dit vlak te volgen en te bekijken welke gevolgen dit heeft voor Gradle build-scripts.



Koog

ASSESS

JetBrains heeft met **Koog** [<https://docs.koog.ai/>] een opensource-framework gelanceerd, dat een developer helpt diverse uitdagingen rondom het maken van AI-agents het hoofd te bieden. Met een AI-agent bedoelen we in deze: Een onafhankelijke applicatie die los gedefinieerde input accepteert, dat naar een LLM stuurt ter interpretatie en het in beginsel non-deterministische antwoord als output teruggeeft. Een applicatie, die op deze manier één of meerdere LLM calls doet, kent een aantal uitdagingen op het gebied van voorspelbaarheid en daarmee herbruikbaarheid en betrouwbaarheid. Koog biedt diverse mechanismes om hiermee om te gaan, zoals de mogelijkheid om workflows op te stellen, conversaties te onthouden, historie te comprimeren en zodoende een kennisbank op te bouwen waaruit een AI-agent kan putten i.p.v. telkens een LLM aan te roepen. Daarbij kan het binnen een conversatie naar een andere LLM overstappen, om bijvoorbeeld te zorgen dat een duurdere of tragere LLM alleen wordt aangeroepen als een snellere, goedkopere geen wenselijk antwoord weet te leveren. Koog is gericht op Kotlin en ondersteunt o.a. OpenTelemetry, Model Context Protocol (MCP), Spring Boot en Ktor.

JDriven ziet in Koog een framework dat helpt de grote risico's van onvoorspelbare uitkomsten en mogelijk hoge kosten van een LLM te mitigeren tot een punt waar je je kunt afvragen, of een applicatie met een dusdanig veilig ingebakerd AI-component niet simpeler te bouwen zou zijn zonder dat AI-component. Maar juist daar, waar developers nog terughoudend zijn om AI in hun applicaties te introduceren en de brug zoeken naar een solution architecture waarin non-deterministische componenten tot hun recht komen, is die veiligheid nodig. We verwachten dat Koog daar een rol van betekenis kan spelen, al moet het daarbij concurreren met LangChain4j en Spring AI.

Model Context Protocol (MCP)

ASSESS

Model Context Protocol (MCP) [<https://modelcontextprotocol.io/introduction>] is een open standaard die applicaties in staat stelt om op een consistente manier context te bieden aan large language models (LLM's). MCP heeft integraties met databases (PostgreSQL, SQLite), platforms, bestandssystemen en vele andere systemen. Door de afhankelijkheid van propriëtaire oplossingen te verminderen, verbetert MCP de interoperabiliteit met meerdere LLM-modellen. Het heeft ook de potentie om niet-technische gebruikers te helpen bij het opvragen en begrijpen van hun data door AI-modellen dynamisch toegang te geven tot relevante context.

De adoptie hangt echter af van industriebrede ondersteuning, en er blijven uitdagingen bestaan bij het standaardiseren van contextoverdracht over diverse systemen, terwijl ook beveiligings- en prestatievraagstukken moeten worden aangepakt. We plaatsen MCP in Assess, omdat het veel potentie toont, maar dit zich nog wel in de praktijk moet bewijzen.

Technieken

Trial

1. Passkeys

3. Embrace Shadow IT

Assess

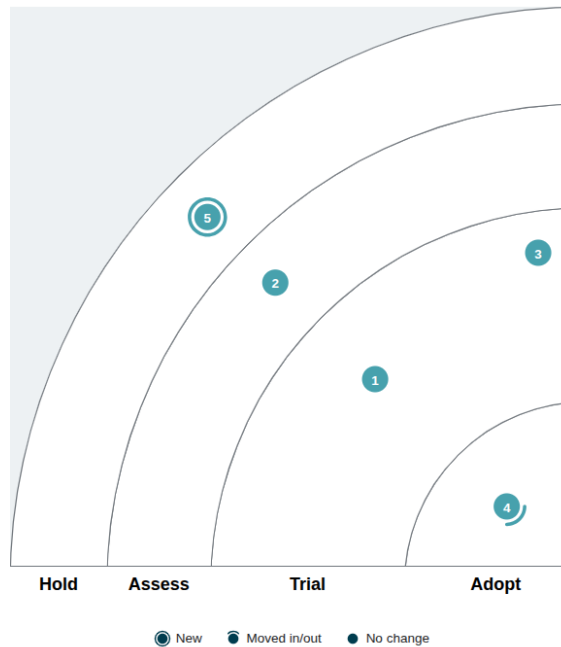
2. AI Agents

Adopt

4. Event Modeling

Hold

5. Vibe coding



Embrace Shadow IT

TRIAL

Shadow IT is een fenomeen waarbij software developers binnen een organisatie IT-faciliteiten optuigen buiten officieel toezicht en regulering door de verantwoordelijke IT-supportafdeling. Dit gebeurt typisch wanneer IT-support er niet voldoende in slaagt om de behoeften van de developers te faciliteren voor het leveren van software. Dat kan liggen in het ongeoorloofd in gebruik nemen van network, storage & compute resources voor te draaien applicaties of voor CI/CD-tooling voor het uitrollen daarvan. Soms komt het tot uiting in zelfgebouwde tooling die buiten audit trails valt, of zelfs in onder de radar in gebruik genomen third party softwarepakketten die via het publieke internet benaderd worden. Shadow IT kan leiden tot een versnipperd softwarelandschap met blinde vlekken. Onderhoud, security en compliance van cruciale onderdelen kan onderbelicht zijn, en management beschouwt shadow IT daarom doorgaans terecht als een risico.

We zien een trend waar platformteams, geboren uit eerdere DevOps-inzichten, gaandeweg weer ondergebracht worden in eigen afdelingen, en weer gestuurd worden op het stabiel en voorspelbaar houden van infrastructuur en CI/CD-tooling, en een overeenkomstige toename van shadow IT door software developers. JDriven moedigt organisaties aan om shadow IT - waar mogelijk, vandaar *Trial* op de Tech Radar - niet krampachtig te beteugelen, en juist de verbinding te zoeken. Developers moeten inzicht hebben in de redenen waarom IT-support grenzen stelt aan wat het faciliteert, en IT-support moet de behoeften van developers kennen voor het groeiende landschap en de eisen aan CI/CD-tooling om software regelmatig te verbeteren. Zoals DevOps het antwoord was op de wrijving ontstaan in een softwarelandschap waarin developers agile software moeten leveren maar wat operations zo stabiel en controleerbaar mogelijk wil houden, zo is shadow IT aan te grijpen voor gecontroleerde innovatie en blijvende effectiviteit van je IT-support. Het omarmen van shadow IT moet dan ook onderdeel zijn van de DevOps-mentaliteit.



Passkeys

TRIAL

Hoewel we niet volledig overtuigd zijn dat passkeys inmiddels een algemeen geaccepteerd begrip zijn, hebben we besloten om deze techniek in *Trial* te plaatsen. De passkey-techniek is wat ons betreft klaar om als alternatief aangeboden te worden naast traditionele authenticatietechnieken, zoals wachtwoorden. Het idee is dat passkeys veilig zijn door het "iets dat je hebt" principe van de "authentication factors" (https://en.wikipedia.org/wiki/Authentication#Authentication_factors), omdat de private key altijd op het apparaat van de gebruiker blijft. Ten opzichte van wachtwoorden voorkomt dit een aantal kwetsbaarheden, die vanwege de menselijke factor vaak kwetsbaar zijn door bijvoorbeeld lage wachtwoordsterkte, hergebruik, datalekken, social engineering en onvoorzichtigheid. Door adoptie op apparaten van grote IT-bedrijven zoals Apple en Google is het gebruiksgemak voor de gemiddelde gebruiker inmiddels zo hoog dat passkeys een realistische optie voor primaire beveiliging zijn geworden.

Mocht je als ontwikkelaar het gebruik van passkeys overwegen, heeft Hanko.io een handige website die het nut van passkeys demonstreert: <https://www.passkeys.io/>.

Denk uiteraard goed na over het proces van accountherstel als je passkeys wilt adopteren. Geef gebruikers voldoende gelegenheid om met vertrouwde processen hun authenticatie aan te passen.

AI Agents

ASSESS

LLM's maken interactie tussen mens en machine mogelijk op een niveau dat we voorheen alleen zagen in films. De mogelijkheid om een vraag te stellen of een taak te definiëren via "natuurlijke" taal is misschien wel de grootste innovatie sinds het internet. Toch communiceren LLM's niet direct met de echte wereld. Ze verwerken een input en produceren een output, niets meer. Moderne **AI Agents** [<https://www.ibm.com/think/topics/ai-agents>] zijn programma's die LLM's gebruiken om zelfstandig te redeneren welke acties nodig zijn om hun taak te voltooien. Ze zijn niet gebonden aan een specifiek script, wat hen in staat stelt om zich dynamisch aan te passen, bijvoorbeeld op basis van het resultaat van een eerdere actie.

Een digitale assistent voor een webshop is een concreet voorbeeld van een agent die je misschien zelf weleens hebt gebruikt. Als een gebruiker wil weten hoe het staat met zijn bestelling, kan hij de assistent simpelweg vragen: "Wat is de status van mijn bestelling?" en hoeft hij geen specifieke formulieren in te vullen. De agent kan vervolgens afleiden dat hij, om deze vraag te beantwoorden, informatie uit de database moet ophalen, maar ook dat hij hiervoor eerst aan de gebruiker het nummer van zijn bestelling moet vragen.

Tegelijkertijd staan AI-agents nog in de kinderschoenen. De onderliggende modellen zijn krachtig, maar foutgevoelig. Betrouwbare inzet vereist aandacht voor robuustheid, beveiliging en ethiek. Toch is de potentie in praktische toepassingen nu al groot genoeg om actief mee te experimenteren — daarom plaatsen we AI-agents in de Assess ring.

Event Modeling

ADOPT

Event Modeling [<https://eventmodeling.org/>] is een methode waarbij het beschrijven van gebeurtenissen in een businessproces centraal staat. Deze gebeurtenissen leiden weer tot het veranderen van informatie.

Het centraal stellen van een businessproces en deze op een eenduidige manier beschrijven, helpt zowel de business als de architecten en ontwikkelaars om een probleem en oplossing te doorgronden. Bij Event Modeling wordt ook de gebruikerservaring meegenomen. Zo ontstaat een compleet beeld van interacties van gebruikers met het systeem en hoe de informatie door het systeem vloeit.

Bij JDriven zien we dat Event Modeling een goede aanvulling op of vervanging van Big Picture Event Storming is, om vanuit een gedeeld begrip van het businessdomein sneller tot concrete implementaties ervan te komen.

Vibe coding

HOLD

Vibe coding is een term die recent een opmars heeft gemaakt in de softwareontwikkeling. Het verwijst naar een ontwikkelmethodiek waarbij programmeurs primair vertrouwen op intuïtie en gevoel bij het iteratief gebruik van AI-tools, met name large language models (LLM's). In plaats van gestructureerde ontwikkelprocessen, draait het om het herhaaldelijk aanpassen van prompts en experimenteren met AI-gegenereerde code totdat het resultaat "goed voelt" en/of lijkt te functioneren.

De praktijk kenmerkt zich door een cyclus van prompt-engineering, waarbij ontwikkelaars instructies verfijnen op basis van de output, zonder noodzakelijkerwijs diepgaand begrip van de onderliggende code, de taal, het framework of de interactie met de infrastructuur waarop het draait. Volgens voorstanders draagt dit bij aan snellere prototyping, terwijl critici wijzen op risico's rondom codekwaliteit, performance, onderhoudbaarheid en begrip van de software.

JDriven schaaft zich achter de critici en plaatst vibe coding op *Hold*. Het betreft een trial-and-error benadering die haaks staat op professionele softwareontwikkeling, waarbij gestructureerd werken, clean code en kwaliteitscontrole centraal staan. Dit gebrek aan systematiek leidt tot "technical debt", beveiligingsproblemen en slecht onderhoudbare codebases. Het feit is dat software development meer behelst dan 'code kloppen' om tot een degelijk resultaat te komen, waar vibe coding geen oplossing voor biedt.

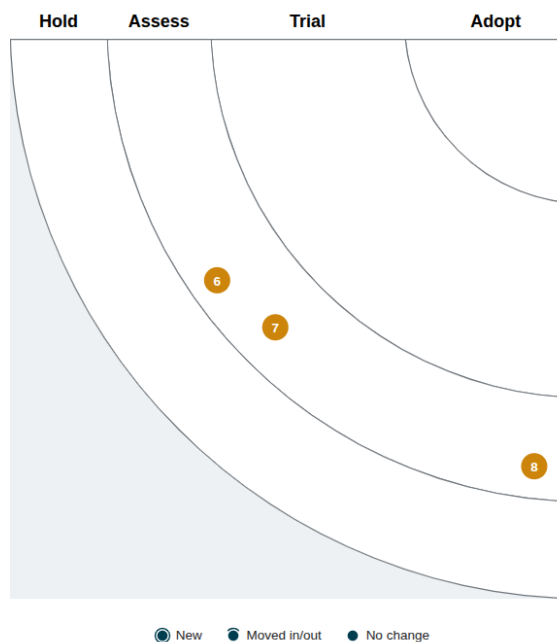
Daartegenover staat dat bewust gebruik van LLM's in een professionele ontwikkelomgeving waarde kan toevoegen. Als onderdeel van een gestructureerd ontwikkelproces kunnen AI-tools uitstekend functioneren voor code review, documentatie, het genereren van boilerplate-code, het verkennen van design patterns en het bieden van enige houvast bij technische foutmeldingen. Het verschil ligt in verantwoorde, methodische integratie versus het ad hoc experimenteren dat kenmerkend is voor vibe coding.



Platforms

Assess

6. EU Cloud alternatives	▼
7. AWS European Sovereign Cloud	▼
8. Proxmox	▼



Dependency on US services

De EU stelt via General Data Protection Regulation (GDPR) eisen aan bedrijven die data, gerelateerd aan inwoners van de EU, verwerken en verzamelen. De Amerikaanse wet FISA702 en Executive Order 12.333 stellen de Amerikaanse overheid echter in staat een Amerikaans bedrijf, of een internationaal bedrijf met een kantoor in Amerika, te dwingen om de GDPR te negeren en inzage in data te geven, zelfs als de betreffende servers buiten de VS liggen. Een additionele overeenkomst tussen de EU en de VS, TADPF, is daarom opgesteld, die in de VS wordt bewaakt door het Privacy and Civil Liberties Oversight Board (PCLOB). Recente grillen van de Amerikaanse politiek omtrent het PCLOB doen diverse Europese bedrijven vrezen, dat dit onvoldoende is om te borgen dat hun data en diensten veilig zijn bij Amerikaanse service providers.

AWS European Sovereign Cloud

ASSESS

Om zorgen van Europese bedrijven rondom PCLOB en GDPR te adresseren, heeft AWS de [AWS European Sovereign Cloud](https://aws.amazon.com/compliance/europe-digital-sovereignty/) [https://aws.amazon.com/compliance/europe-digital-sovereignty/] aangekondigd, wat een AWS Cloud is die geheel in de EU gehuisvest moet zijn, bemand en bediend door inwoners van de EU. De eerste locatie van de AWS European Sovereign Cloud zal in Duitsland geplaatst worden en is gepland voor eind 2025. AWS is via een FAQ en whitepapers op de site van deze nieuwe service transparant omtrent het willen naleven van de GDPR en over de impact van FISA702, en benadrukt dat ieder verzoek van een overheid om data in te zien kritisch bekeken zal worden.

Voor een bedrijf dat voor zijn software hevig heeft geïnvesteerd in AWS-kennis en -implementatie, en op zoek is naar een manier om GDPR compliant te blijven zonder die investeringen af te schrijven, denkt JDriven dat het de moeite waard is om te analyseren of deze AWS European Sovereign Cloud een veilig alternatief is. Daarbij moet opgemerkt worden, dat de AWS European Sovereign Cloud aanzienlijk minder services biedt dan de reguliere AWS-cloud. Wat er bovendien in de voorwaarden te lezen valt, is dat AWS uiteindelijk toch zal moeten voldoen aan wettelijke verzoeken om data in de AWS European Sovereign Cloud in te zien, wat de vraag doet rijzen of het uiteindelijk uitmaakt of medewerkers en servers Europees zijn, zolang het moederbedrijf AWS Amerikaans is en dus aan die wetten is gebonden. Dit alles maakt, dat we dit als Assess op de radar plaatsen.

EU Cloud alternatives

ASSESS

Meer dan een decennium geleden besloten diverse Amerikaanse bedrijven als Amazon, Google en Microsoft, dat ze hun serverparken konden uitbreiden en verhuren aan andere bedrijven. Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) en Software-as-a-Service (SaaS): "de cloud" was geboren. Over de jaren heen breidden cloudleveranciers hun diensten uit en kwamen er "cloud-native" services bij: als klant installeer je bijvoorbeeld niet meer een database op een gehuurde server, maar neem je een databaseservice af, waarbij de cloudleverancier ook installatie en onderhoud van de software voor zijn rekening neemt. Dit heeft organisaties geholpen om kostenefficiënt infrastructuur schaalbaar te maken en heeft ze bovendien in staat gesteld zich te concentreren op hun eigen unieke competitieve kerncompetenties. Softwareconsultants waarschuwen echter ook al tijden voor vendor lock-in: cloudleveranciers willen klanten lokken met een uniek aanbod, maar dat maakt dat een klant ook lastiger weg kan migreren van een cloud, als ze die services gebruiken.

De laatste paar maanden wordt duidelijk dat er een additionele overweging te maken is rondom vendor lock-in. Amerikaanse cloudleveranciers moeten zich schikken naar een regering die zich grillig of zelfs agressief opstelt richting het buitenland, en daarbij een handelsoorlog heeft ontketend met Europese landen. Europese organisaties moeten zich daarom afvragen hoe kwetsbaar ze zijn als ze hun software en data onderbrengen bij Amerikaanse cloudleveranciers. Cloud-agnostische platforms zoals Kubernetes en Open Source Software (OSS) kunnen helpen met ontkoppelen van de softwarearchitectuur en de onderliggende infrastructuur en zodoende het risico enigszins afdekken. Maar JDriven denkt dat het ook tijd is om te kijken naar de diverse Europese cloudleveranciers en te overwegen in hoeverre hun cloudbaanbod voldoende aansluit op hun software- en infrastructuurbehoeftes. Bekendere spelers op de Europese markt zijn bijvoorbeeld Cyso cloud, OVH Cloud, Exoscale, Hetzner, Scaleway, T-Systems, en Proact. Ook is er het Europese Gaia-X initiatief, waar diverse cloudleveranciers zich bij aangesloten hebben t.b.v. interoperabiliteit en gedecentraliseerde dataveiligheid bij het gebruik van meerdere clouds. Een voorbeeld van een Europees SaaS-initiatief is de Duitse deelstaat van Sleeswijk-Holstein, die digitale soevereiniteit nastreeft via migratie naar een open source IT-infrastructuur gebaseerd op Nextcloud Hub.

Proxmox

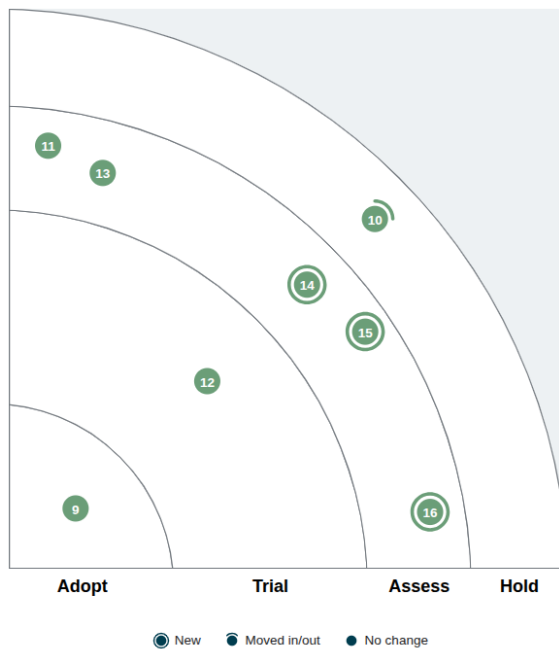
ASSESS

Proxmox [<https://www.proxmox.com/en/>] biedt een alternatief voor het veelgebruikte VMWare-platform. De Proxmox Virtual Environment (PVE) is een opensource-virtualisatieplatform voor het draaien van containers en/of virtual machines met verschillende operating systems. De PVE ondersteunt high availability over meerdere servers, en heeft efficiënte backupfunctionaliteit.

Ondanks dat het geen nieuwe techniek is, is het in de context van groeiende onzekerheid rondom internationale clouddiensten belangrijk om het zelf hosten en in beheer nemen van IT-diensten op eigen servers opnieuw te overwegen. Wij zien Proxmox als een interessantere optie dan VMWare, dat sinds de overname door Broadcom almaar stijgende licentiekosten kent. We hebben Proxmox in Assess gezet omdat de toepasbaarheid volledig afhankelijk is van de situatie. De afwegingen zijn lagere (licentie-)kosten en meer controle, tegenover het verlies aan schaalbaarheid en beschikbare support.



Tools



Adopt

9. IntelliJ HTTP client

Hold

10. JetBrains AI

Assess

11. DeepSeek

13. Cursor IDE

14. Maven 4

15. Anubis

16. BlueDolphin

Trial

12. Nushell

IntelliJ HTTP client

ADOPT

De IntelliJ HTTP-Client is een geïntegreerde functionaliteit binnen IntelliJ IDEA die ontwikkelaars in staat stelt om HTTP-verzoeken direct vanuit de IDE op te stellen en uit te voeren. Deze client maakt gebruik van `.http` of `.rest` bestanden, waarin http requests eenvoudig kunnen worden gedefinieerd. Dit maakt het mogelijk om API's snel en efficiënt te testen zonder afhankelijk te zijn van externe tools zoals Postman.

De HTTP-Client ondersteunt het gebruik van variabelen, scripting en diverse vormen van authenticatie, wat het geschikt maakt voor zowel eenvoudige als geavanceerde use-cases. Responses worden overzichtelijk weergegeven en opgeslagen naast je request bestanden, waardoor het analyseren van deze responses eenvoudig is.

Een belangrijk voordeel is dat deze request-bestanden kunnen worden opgeslagen in de projectstructuur, naast de broncode. Dit bevordert herbruikbaarheid, versiebeheer en samenwerking binnen teams, en draagt bij aan een beter gedocumenteerde en reproduceerbare API-ontwikkeling. Omdat IntelliJ een veel gebruikte IDE is, is het ook een logische keuze voor ontwikkelaars die al vertrouwd zijn met de omgeving, en niet nog een aparte tool willen gebruiken voor API-testing.

De client bestaat al enige tijd, maar er is recent veel goed werkende functionaliteit toegevoegd, waaronder het gebruik van environment-files, voor het veilig gebruik van authenticatie, OAuth2-flows en het gebruik van variabelen. Voor ontwikkelaars die al met IntelliJ werken, biedt het alle features die nodig zijn om API's te testen en te ontwikkelen, zonder dat ze daarvoor naar een aparte tool dienen te switchen.

Nushell

TRIAL

Nushell [<https://www.nushell.sh>] is een nieuw type shell. Het doel van Nushell is om de Unix-filosofie van shells, waarbij pipes eenvoudige commando's met elkaar verbinden, te combineren met een krachtige programmeertaal. Zo wordt Nushell zowel een shell als een programmeertaal in één pakket, dat draait op Linux, macOS en Windows. Nushell combineert een volledige shell met een uitgebreide programmeertaal. Code geschreven met Nushell kan worden uitgevoerd op Linux, MacOS en Windows. Nushell gebruikt overal datastructuren, bijvoorbeeld de output van een `ls` commando, maar ook de output van een REST API call. Er zijn standaard commando's die gecombineerd kunnen worden voor het vinden, sorteren, filteren en converteren van data. Bijvoorbeeld om bestanden te vinden die groter zijn dan 10 MB, en vervolgens te sorteren op de laatste wijziging: `ls | where size >= 10mb | sort-by modified`.

Nushell lijkt al langer te bestaan, maar het is nog wel een 0.x versie. Daarom is het wel interessant om Nushell te onderzoeken en uit te proberen, om te kijken wat de voor- en nadelen zijn ten opzichte van bestaande shells en tools.

Anubis

ASSESS

Anubis is een softwareprogramma dat een op JavaScript gebaseerde proof-of-workuitdaging aan websites toevoegt voordat gebruikers deze kunnen bezoeken. In principe kunnen alleen moderne browsers die proof-of-workuitdaging uitvoeren en fungeert dit dus als een alternatief voor CAPTCHA's, waar geen gebruikersinteractie voor nodig is. Het is ontworpen om het internet te beschermen tegen de eindeloze stroom requests van AI-bedrijven. Een scraperbot kan standaard geen proof-of-workuitdaging uitvoeren en dus geen zinvolle verbinding opzetten. Anubis fungeert zodoende als een zogenaamde 'tar pit': het remt met name scrapers af, die data zoeken voor AI-training.

Anubis is een jong project en het is afwachten of deze aanpak standhoudt; scrapers kunnen erop aangepast worden. Maar het aanpassen van een scraper, zodanig dat hij de proof-of-workuitdaging wel kan verrichten, zou hem langzamer maken, waardoor hij minder requests kan versturen. Omdat het toepassen van Anubis een relatief kleine investering is, achten we het de moeite waard om te proberen. Van belang is om Anubis te configureren dat bots van bijvoorbeeld zoekmachines doorgelaten worden, indien gewenst. De standaard-afbeelding van Anubis, die bezoekers te zien krijgen bij een bezoek, zal niet bij het thema van alle websites passen, maar die kan aangepast worden.



BlueDolphin

ASSESS

BlueDolphin [<https://www.valueblue.com/bluedolphin>] is een platform voor enterprise-architectuur en procesmodellering. BlueDolphin biedt een set aan functionaliteiten voor het visualiseren van bedrijfsstructuren, applicatielandschappen en informatiestromen. De kracht van BlueDolphin zit in het ondersteunen van samenwerking tussen verschillende rollen binnen een organisatie en stelt teams in staat om zowel de huidige als de gewenste architectuur visueel en gestructureerd vorm te geven. Dit vergroot het inzicht in de impact van veranderingen en helpt bij het onderbouwen van architectuurkeuzes.

Wel brengt de veelheid aan mogelijkheden ook een keerzijde met zich mee. Door de grote hoeveelheid opties en configuratiemogelijkheden is het voor nieuwe gebruikers, en soms zelfs voor ervaren gebruikers, makkelijk om de weg kwijt te raken. De leercurve is steil, en zonder duidelijke richtlijnen of governance kan het gebruik van BlueDolphin snel ongestructureerd of inconsistent worden.

Hoewel BlueDolphin potentie heeft om organisaties te helpen bij het structureren en communiceren van hun architectuur, vereist het een duidelijke visie en begeleiding om effectief ingezet te worden. Het is daarom belangrijk om vooraf goed na te denken over het doel, scope en de opleidingsbehoeftes van het gebruik binnen de organisatie.

Daarom wordt BlueDolphin in het kwadrant Assess geplaatst van deze editie van onze Tech Radar.

Cursor IDE

ASSESS

Cursor IDE [<https://www.cursor.com/>] is een AI code-editor die gestoeld is op Visual Studio Code (VS Code). Het biedt geïntegreerde AI waarbij het gebruik van AI-support in het ontwikkelproces ingebakken zit. Verder bevat het functies zoals code completion, code generatie, code review, etc.

We zien Cursor recentelijk regelmatig gebruikt worden door sprekers op conferenties, en het is een interessant alternatief voor VS Code. Cursor onderscheidt zich door de geïntegreerde AI-support ten opzichte van AI-plugins die je in andere IDE's kunt gebruiken. Vanuit JDriven zien wij voordelen, zoals het beter meenemen van AI in de developerworkflow en de mogelijke toename in productiviteit doordat met AI randzaken sneller opgezet kunnen worden. Wel merken we dat er een leercurve is om er efficiënt mee te werken. Bovendien zien we in het algemeen dat zgn. *vibe coding* ertoe kan leiden dat een developer meer bezig is met het reviewen en corrigeren van door AI gegenereerde code, dan met het zelf uitdenken en begrijpen van oplossingen. Dit is een risico, omdat ontwikkelaars zelf verantwoordelijk blijven voor de code die op productie staat. Het raadplegen van de LLM's werkt met credits en wachttijden, die omzeild kunnen worden door te betalen per request. Het is belangrijk te weten, dat requests naar Cursor servers worden gestuurd, wat privacy- en beveiligingsrisico's met zich meebrengt. Deze voor- en nadelen doen ons besluiten Cursor IDE op Assess te plaatsen.

DeepSeek

ASSESS

DeepSeek [<https://tweakers.net/nieuws/231566/nederlandse-ambtenaren-mogen-deepseek-niet-gebruiken-van-kabinet.html>] is een alternatief LLM voor ChatGPT dat kan worden gebruikt voor verschillende taken, waaronder code-assistentie. Het redeneermodel en de training stellen het in staat om nauwkeurig te presteren, zelfs met minder resources, waardoor het deel uitmaakt van een nieuwe generatie efficiënte open-source LLM's die lokaal kunnen draaien en toch sterke prestaties leveren.

Ondanks zijn potentieel is DeepSeek niet zonder controverse. De rechtmatigheid van de data-acquisitie roept, net als bij ChatGPT, vragen op, en er zijn zorgen over mogelijke Chinese censuur. We hebben gemerkt dat veel van onze klanten de voorkeur geven aan het vermijden van cloud-gebaseerde LLM's vanwege het risico dat hun gegevens worden gedeeld. Een lokaal draaiende LLM zoals DeepSeek pakt dit probleem aan door ondersteuning te bieden zonder informatie extern te versturen.

Wanneer het wordt gecombineerd met tools zoals Devox Genie, kan DeepSeek in-IDE suggesties bieden voor code. Zijn mogelijkheden reiken echter verder dan alleen coderen, waardoor het een veelzijdige AI-tool is.

We plaatsen DeepSeek in Assess, omdat het lokaal kan worden gebruikt en een interessant alternatief biedt, maar nog verder onderzoek vereist.

Maven 4

ASSESS

Maven 4 [<https://maven.apache.org/whatsnewinmaven4.html>] bevindt zich momenteel in Release Candidate-stadium en brengt significante verbeteringen voor Java-projecten. De belangrijkste winst zit in de prestaties: parallelle dependency resolution en verbeterde caching kunnen build-tijden significant verkorten, vooral merkbaar bij grote multimodule-projecten. Daarnaast introduceert Maven 4 de scheiding tussen build en consumer POM's, waardoor ontwikkelaars flexibeler configuraties kunnen gebruiken zonder afhankelijke projecten te beïnvloeden.

De verbeterde foutmeldingen en automatische parentversie-inferentie maken het werken met Maven intuïtiever. Voor teams die werken met moderne CI/CD-pipelines zijn de betere integratiemogelijkheden met container-based builds en cloud-native omgevingen een welkome toevoeging. Maven 4 introduceert een nieuw dedicated bom packaging-type met ondersteuning voor exclusions en classifiers, wat dependency management in multi-module projecten vereenvoudigt. De langverwachte ingebouwde ondersteuning voor CI-friendly versioning met `${revision}+`, `${sha1}+` en `${changelist}+` properties werkt nu zonder de Flatten Maven Plugin, samen met automatische parent- en subproject-versionering.

De lifecycle is veranderd van een graph- naar een tree-structuur met nieuwe phase types zoals `before:all` en `after:all` voor meer granulaire controle. Profile management is flexibeler geworden met optionele profiles die een `?` prefix gebruiken, waardoor builds niet meer breken als een profile niet bestaat. Deze verbeteringen maken Maven 4 significant gebruiksvriendelijker voor moderne development workflows. Er zijn wel enkele breaking changes, met name rond plugin-compatibiliteit en de strengere POM-validatie, waardoor een zorgvuldige migratie nodig is. Merk ook op dat Java 17 de minimale vereiste is voor Maven 4.

We verwachten dat wanneer Maven 4 GA (General Availability) bereikt, het een solide keuze zal zijn voor projecten, gezien de significante prestatieverbeteringen en de verbeterde developer experience. Daarom is het zaak nu al te kijken naar wat er bij een migratie komt kijken. Gezien de significante prestatieverbeteringen en de verbeterde developer experience plaatsen wij Maven 4 in de Assess ring.



JetBrains AI

HOLD

JetBrains [<https://www.jetbrains.com/ai/>] heeft twee AI-technologieën geïntroduceerd: Mellow (oktober 2024) en Junie (januari 2025). Mellow is een gespecialiseerde LLM, geoptimaliseerd voor code-completion met ondersteuning voor Java, Kotlin, Python en Go. Het heeft een drie keer snellere responstijd en 40% acceptatiegraad van de suggesties. Junie is een AI-coding agent binnen de JetBrains Pro- en Ultimate-lijn die 53,6% (volgens SWEbench) van de ontwikkelingstaken zelfstandig kan oplossen. Waar Mellow functioneert als intelligente code-aanvuller, is Junie een semi-autonome assistent die complexere taken voltooit zoals het genereren van code, het uitvoeren van inspecties en het schrijven van tests. Beide tools zijn ontworpen met privacy als prioriteit – Mellow is getraind op publiek beschikbare code met permissieve licenties – en geven ontwikkelaars controle over voorgestelde wijzigingen. JetBrains' visie met deze tools is om softwareontwikkeling productiever en plezieriger te maken door directe assistentie (Mellow) te combineren met een AI-partner die de ontwikkelingsworkflow fundamenteel kan veranderen (Junie).

Voor een ontwikkelaar is vaak het verzinnen en uitwerken van oplossingen het plezierige deel van ons werk. We besteden al veel tijd aan het lezen en begrijpen van code die anderen (inclusief AI) hebben geschreven, en niet actief betrokken zijn bij het zelf uitwerken van oplossingen maakt dat werk moeilijker. Het is belangrijk om te evalueren of deze technologieën daadwerkelijk de juiste problemen aanpakken, zoals het reduceren van boilerplate code en het bieden van een efficiënt alternatief voor Stack Overflow. JetBrains heeft de ondersteuning voor het lokaal draaien van Mellow in de AI-assistent sterk verbeterd (middels Ollama of LM Studio), waardoor codeverkeer naar de JetBrains-servers voorkomen kan worden. Dit adresseert een potentiële dealbreaker voor organisaties met strikte privacy-eisen. Bovendien is het goed om te onthouden dat JetBrains AI meer omvat dan alleen deze twee technologieën. Ondanks de geavanceerde functionaliteit van Junie als autonome agent en de sterke privacyfocus door lokale LLM-ondersteuning, zien we binnen onze organisatie vrijwel geen adoptie van JetBrains' eigen AI-tools. De ontwikkelaarsfocus ligt veel meer op bekendere, platform-agnostische alternatieven die beter aan lijken te sluiten bij de dagelijkse workflow, zoals GitHub Copilot, Cody en Claude Code. Aangezien adoptiegraad een belangrijke factor is voor het plaatsen binnen een ring, hebben wij Mellow en Junie in de *Hold* ring geplaatst, en moedigen aan om meerdere tools zelf te evalueren om te bepalen welke de meeste waarde biedt.





Commit.
Develop.
Share.