

talents décryptage

En générant du code, en le complétant ou en l'optimisant, l'IA générative réduit les temps de développement et la dette technique. Si les développeurs se sont largement emparés de ces outils, ils conservent une distance critique.



Gains de productivité des développeurs avec l'IA générative : jusqu'où ?

L'IA générative ne sert pas qu'à créer du texte, des images ou des vidéos. Elle peut aussi générer du code, compléter un programme partiellement écrit, traduire un code d'un langage de programmation à un autre. Un modèle d'IA permet non seulement de faire gagner du temps aux développeurs, en éliminant les tâches répétitives et fastidieuses, mais aussi d'améliorer la qualité du code produit. En procédant à une revue de code automatisée, une IA fait la chasse aux bugs tout en suggérant des pistes d'optimisation. Enfin, elle peut documenter le programme finalisé.

Dans ce domaine, GitHub Copilot, basé sur Codex d'OpenAI, a longtemps fait la course en tête. L'assistant de la filiale de Microsoft a depuis été rattrapé par la patrouille. Il ne se passe pas une semaine sans qu'un éditeur n'annonce avoir intégré l'IA générative à son portefeuille.

Principal concurrent de GitHub,

GitLab propose ainsi dans la version 16 de sa plateforme DevSecOps un ensemble de fonctionnalités couvrant tout le cycle de vie du code, de l'écriture assistée de code à la détection proactive des vulnérabilités en passant par les tests automatisés.

Avec Viya Workbench, SAS propose de son côté un environnement de développement pour construire des modèles « prêts pour la production » en Python, R ou SAS. Salesforce a lui ajouté un copilote à Einstein, son IA maison. Einstein Copilot vise à transformer des requêtes en langage naturel en code Apex, à suggérer du code plus performant, et à rechercher de manière proactive les éventuelles vulnérabilités.

Des « copilotes » comme s'il en pleuvait

Les éditeurs manquant décidément d'originalité, Appian a aussi nommé son assistant AI Copilot. Il vient simplifier le développement de logiciels

sur sa plateforme low-code. Son concurrent OutSystems a quant à lui baptisé AI Mentor sa solution de développement, de sécurité et d'analyse de la qualité basée sur l'IA.

Comment est ressentie cette évolution des pratiques chez les principaux intéressés ? Pour répondre à cette question, le site Stack Overflow a interrogé en ligne 90000 développeurs du monde entier. Publiée en juin, l'enquête montre que l'adoption est déjà massive. 44% d'entre eux ont déjà intégré l'IA dans leurs processus de développement et 26% prévoient de le faire prochainement. La grande majorité des développeurs recourent à l'intelligence artificielle pour écrire du code (83%) et pour le déboguer (49%). Les deux outils les plus massivement retenus sont ChatGPT (83%) et GitHub Copilot (56%).

Si la curiosité est de mise, les développeurs accordent – pour l'heure – une confiance mesurée à l'IA. Seuls 2,9% des répondants affirment faire pleinement confiance à la précision de ces outils et le double (5,5%) ne lui accordent aucun crédit. Entre les deux, une forte proportion (39,3%) fait « plutôt » confiance en l'IA. La perception diffère selon le pays d'origine. Le niveau de confiance est plus élevé en Inde (55,2%), au Brésil (45%) ou aux États-Unis (37,5%) qu'en France (36,1%). Pour expliquer cette tendance, l'étude met en avant le cadre réglementaire européen (RGPD et futur AI Act) plus restrictif.

Les développeurs blockchain, les data scientists et les développeurs front end ou full stack sont plus enclins à recourir à l'IA. À l'inverse, ceux œuvrant dans le logiciel embarqué ou le hardware en tirent peu profit. En ce qui concerne les bénéfices constatés, l'augmentation de la productivité (32,8%) arrive largement en tête. L'IA peut aussi servir de tuteur pour apprendre un nouveau langage ou gagner en qualité. En revanche, seuls 3,8% des développeurs pensent qu'elle peut améliorer le travail collaboratif. L'IA pourrait plutôt mettre à mal certaines pratiques comme la revue de code par les pairs.

XAVIER BISEUL