

Code et IA Gen : les 4 plus grands défis que le dernier rapport du Gartner omet de mentionner

Selon l'analyste, l'IA exigera de 80 % des développeurs qu'ils se perfectionnent. Mais ce n'est pas tout. De la génération de mauvais code aux révisions sans fin, voici ce qui est absent du rapport.

Oui, le Gartner est l'un des analystes les plus respectés de l'industrie de la tech. Son activité consiste à fournir des services de conseil personnalisés, à organiser des conférences et à publier des rapports d'analyse sectorielle multi-clients

Devenir client de Gartner n'est pas donné. Les premiers prix sont à 30 000 \$ par an. La façon dont Gartner (et, pour être juste, la plupart des autres groupes d'analystes) suscite l'intérêt est de publier des communiqués de presse qui résument les conclusions de ses rapports de recherche, dans l'espoir qu'ils soient repris par ZDNET par exemple.

Et comme ces communiqués de presse contiennent souvent des éléments qui expliquent les tendances du secteur, nous nous y référons souvent pour essayer de comprendre l'état de l'industrie

Gartner est donc de retour avec un nouveau communiqué de presse (en annexe) . Celui-ci est intitulé "Gartner Says Generative AI will Require 80% of Engineering Workforce to Upskill Through 2027" (L'IA générative exigera une mise à niveau de 80 % de la main-d'œuvre en ingénierie d'ici 2027).

Le communiqué de presse présente ensuite les prévisions de Gartner concernant l'impact à court, moyen et long terme de l'IA sur l'ingénierie logicielle.

Dans cet article, j'examinerai donc les conclusions de Gartner. Mais aussi plusieurs omissions flagrantes dans les conclusions. J'ai contacté Gartner pour demander la possibilité d'examiner le rapport complet, que l'entreprise a eu la gentillesse de me fournir.

Le rapport complet mentionne très brièvement certaines des omissions. Mais beaucoup de gens se baseront sur le communiqué de presse pour se faire une opinion. Plus que sur le rapport complet qui se trouve derrière un pare-feu à cinq chiffres.

Parlons d'abord du titre du rapport

Le titre indique que l'IA exigera que 80 % de la main-d'œuvre technique se perfectionne au cours des deux prochaines années. Plus loin dans le communiqué, Gartner précise qu'il s'agit de l'ingénierie logicielle et non de toutes les disciplines de l'ingénierie.

L'implication est que l'IA va tellement perturber l'ingénierie logicielle que si vous ne vous perfectionnez pas, vous êtes foutu. C'est vrai. Mais ce n'est pas une nouveauté.

L'actualisation des connaissances, l'apprentissage de nouvelles choses et la mise à jour des technologies sont des aspects essentiels du travail de tout ingénieur logiciel. Dans cet article,

j'utiliserai des termes comme programmeur, codeur, ingénieur logiciel et développeur de manière assez interchangeable. Il existe des différences subtiles, mais pour les besoins de notre discussion, ils sont tous applicables.

Ce que je veux dire, c'est que si vous êtes un codeur, vous devez constamment apprendre.

Il n'y a rien de surprenant à devoir se perfectionner en tant qu'ingénieur logiciel. Les programmeurs doivent toujours se perfectionner. Chaque nouvelle version de système d'exploitation, chaque nouveau langage, chaque nouvelle API, chaque nouveau matériel mis sur le marché nécessite une mise à niveau.

L'IA exigera également des ingénieurs qu'ils développent leurs compétences. Mais ce n'est pas un grand changement dans ce qu'il faut faire pour réussir une carrière d'ingénieur logiciel ou de développeur.

Alors oui, bien que le titre soit crédible, il n'est pas aussi important qu'il semble le laisser entendre.

Impact de l'IA à court, moyen et long terme

Voici comment Gartner décrit les étapes de l'impact de l'IA.

- À court terme : "Les outils d'IA généreront des augmentations de productivité modestes. Les avantages de l'IA en termes de productivité seront les plus significatifs pour les développeurs seniors dans les organisations ayant des pratiques d'ingénierie matures".
- À moyen terme : "Les agents d'IA transformeront les modèles de travail des développeurs en leur permettant d'automatiser et de se décharger entièrement de davantage de tâches. Cela marquera l'émergence de l'ingénierie logicielle IA- native, lorsque la plupart des codes seront générés par l'IA plutôt qu'écrits par des humains."
- À long terme : "Si l'IA rendra l'ingénierie plus efficace, les organisations auront besoin d'ingénieurs logiciels encore plus compétents pour répondre à la demande croissante de logiciels alimentés par l'IA."

Je ne suis pas en désaccord avec cette évaluation, bien qu'il ne soit pas possible de répartir ces trois résultats dans le temps. Les outils d'IA, maintenant que nous les avons, seront en mesure d'améliorer le codage à court et à long terme.

Les débogueurs symboliques, par exemple, existent depuis une quarantaine d'années. Et nous les utilisons encore quotidiennement. Ils se sont améliorés, ils font plus de choses et ils reflètent les nuances de leur environnement de travail. Il en sera de même pour la programmation assistée par l'IA.

"Une nouvelle race de professionnels du logiciel, l'ingénieur en IA"

Selon l'analyste de Gartner Philip Walsh, "la création de logiciels par l'IA exigera une nouvelle race de professionnels du logiciel, l'ingénieur en IA".

"L'ingénieur en IA possède une combinaison unique de compétences en génie logiciel, en science des données et en IA/apprentissage machine (ML), des compétences qui sont recherchées."

Gardez à l'esprit qu'il existe une distinction lorsqu'il s'agit de l'impact de l'IA sur le codage.

- Parfois, les codeurs utiliseront l'IA pour programmer (pour faire fonctionner plus de code).
- Parfois, les développeurs utiliseront l'IA dans la programmation (pour rendre le code et la solution sur lesquels ils travaillent plus intelligents).

Gartner a mené une enquête à la fin de l'année dernière auprès de 300 responsables de l'ingénierie logicielle pour connaître les rôles de chacun. 56 % d'entre eux ont déclaré que les ingénieurs en IA et en apprentissage automatique étaient les plus demandés et que l'application de l'IA et de l'apprentissage automatique aux applications était un domaine dans lequel leurs équipes avaient besoin d'accroître leurs compétences.

C'est tout à fait logique. Les fonctions d'IA et d'apprentissage automatique ont le potentiel d'ajouter une valeur considérable aux applications dans presque tous les domaines.

Les 4 omissions flagrantes du rapport du Gartner

Selon Philip Walsh, "à l'ère de l'IA native, les ingénieurs logiciels adopteront un état d'esprit "AI-first", où ils se concentreront principalement sur l'orientation des agents d'IA vers le contexte et les contraintes les plus pertinents pour une tâche donnée."

Le principe de base de la programmation "AI-first" est que les développeurs confient des tâches à une IA et que celle-ci leur renvoie le code du projet. Ce principe est analogue à celui d'un responsable de la programmation qui confie des tâches aux développeurs de l'équipe, et ces derniers lui renvoient le code du projet.

Le rapport complet mentionne que les développeurs seniors peuvent être chargés de vérifier la qualité du code dans les organisations dépourvues d'outils d'examen automatisé du code.

Voici les **quatre omissions du communiqué** de presse de Gartner

- 1 - Il est toujours nécessaire de procéder à des révisions pour que le code soit conforme aux spécifications
- 2 - Les tests et le contrôle de la qualité sont requis pour tous les logiciels
- 3 - Il y a besoin de mises à jour, de maintenance, de corrections de bogues et d'optimisation des performances tout au long du cycle de vie.
- 4 - Il est beaucoup plus difficile de maintenir un code que l'on n'a pas écrit.

Il est impossible de discuter de l'avenir de l'ingénierie logicielle sans accorder une attention particulière à ces quatre aspects du métier.

1. Réviser le code pour répondre aux spécifications

Le scénario est simple. Vous donnez une instruction à une IA ou un ensemble de directives à une équipe de programmation. Le temps passe (moins pour l'IA, plus pour l'équipe) et vous obtenez un morceau de code supposé fonctionner.

Donc vous l'essayez. Et il n'est pas tout à fait correct. Il ne sera jamais parfait du premier coup. Même si vous faites votre propre codea, la première exécution après l'écriture du code comportera des bogues. C'est l'une des lois naturelles de l'univers.

À l'heure actuelle, vous pouvez dire à ChatGPT que ce qu'il vous a donné ne fonctionne pas. Ou lui donner le message d'erreur et il générera un tout nouveau bloc de code. Mais les IA ont beaucoup plus de mal à réviser progressivement leur travail pour s'adapter à ce qui est nécessaire.

Et ce processus s'applique que l'IA soit de texte à texte ou même de texte à image. Essayez de demander à Midjourney d'apporter les modifications que vous souhaitez à une image. Vous pouvez spécifier de petites zones sur lesquelles travailler, mais Midjourney fera alors ce qu'il veut faire.

Pour que les développeurs puissent confier des tâches à une IA et que celle-ci leur renvoie le code du projet, l'IA doit être capable d'accepter les critiques et d'apporter de petites modifications ciblées.

À l'heure actuelle, chaque révision renvoyée est un véritable coup de dés

2. Tests et contrôle de la qualité

Il est possible de faire en sorte qu'une IA teste les résultats d'une autre IA. Je l'ai fait avec d'assez bons résultats. Notamment en vérifiant comment les expressions régulières sont construites et fonctionneront.

Mais compte tenu des hallucinations des IA et de la fréquence à laquelle une IA écrit un code qui ne fonctionne tout simplement pas, des protocoles et des tests devront être développés dans le cadre du processus de programmation avec de l'IA.

Tout cela nécessitera probablement que les humains fassent une grande partie du travail.

Nous ne pourrons pas compter sur l'IA pour coder tant que nous ne serons pas sûrs que le code est bon. Plus le projet est important, plus le code est complexe, plus nous aurons besoin de programmeurs et de gestionnaires de projet pour superviser l'ensemble du processus

3-. Mises à jour, maintenance, corrections de bogues, optimisation des performances

Les logiciels sont loin d'être statiques. Une fois en production, les bogues devront être corrigés, les mises à jour devront être mises en œuvre, les performances devront être améliorées, etc. Aucune de ces tâches ne peut être confiée à une IA pour l'ensemble d'un produit.

Cela ne veut pas dire qu'une IA ne peut pas aider. On peut certainement lui confier une fonction, par exemple, et lui demander de réécrire (notez que je n'ai pas dit "mettre à jour") la fonction pour en améliorer les performances.

Bien sûr, vous avez maintenant une fonction entièrement nouvelle qui doit être soumise à des tests fonctionnels. Si quelque chose ne fonctionne pas lors de la réécriture, vous ne pouvez pas vous contenter des quelques lignes qui ont été modifiées. Vous devez trouver le bogue dans l'ensemble du bloc de code

4. Maintenir le code que vous n'avez pas écrit

Maintenir le code que vous avez écrit est un travail. Et maintenir un code que vous n'avez pas écrit, c'est une tonne de travail. Lorsque vous écrivez le code, vous savez pourquoi le code a été structuré de la sorte.

Lorsque vous maintenez un code écrit par quelqu'un d'autre, ce code est complètement nouveau. Vous devez d'abord déconstruire toute l'architecture du code. Si vous vous y prenez bien, vous documenterez soigneusement tout ce que vous découvrirez. Ce processus de déconstruction est très chronophage. Et ce avant de modifier ou d'ajouter ne serait-ce qu'une ligne de nouveau code.

J'ai une certaine expérience en la matière. En 2015, j'ai adopté 10 plugins WordPress open source. Cela signifiait que j'étais responsable de la maintenance du code, de la correction des bogues et des failles de sécurité. Et, pour certains d'entre eux, de leur transformation en produits plus fonctionnels sur le plan commercial.

Non seulement j'ai dû déconstruire la façon dont les programmeurs avaient écrit leur code. Mais j'ai également découvert que, dans certains cas, je détestais leur style de programmation.

Dans la plupart des cas, le code fonctionnera lorsqu'il est écrit dans différents styles. Mais le cerveau de chaque programmeur fonctionne différemment. Je suis particulièrement pointilleux sur le style du code. S'il est écrit comme je l'aime, je suis incroyablement productif. Mais si le style de codage m'ennuie, ma productivité chute. Et je deviens grincheux.

J'ai essayé de faire convertir par ChatGPT un des styles de codage ennuyeux vers un style que j'aime. ChatGPT n'a pas réussi. Avec le temps, il sera peut-être possible de former soigneusement une IA à un style de codage. Et de lui faire mettre à jour le code. Mais nous n'en sommes pas encore là.

Ainsi, la maintenance du code produit par l'IA sera non seulement plus difficile que l'écriture du code à l'origine. Mais elle pourra également entrer en conflit avec les bonnes pratiques de style. Ce qui rendra le travail encore plus fastidieux.

L'IA remplacera-t-elle les développeurs ?

Le Gartner dit que "l'IA ne remplacera pas les ingénieurs en logiciel (et pourrait, en fait, en nécessiter davantage)". Je suis tout à fait d'accord.

Au-delà de l'utilisation de l'IA pour aider à programmer ou pour rendre les programmes plus performants, il y a beaucoup de travail à faire en ce qui concerne la logistique fondamentale des projets de programmation.

Il ne fait aucun doute que l'IA peut aider à la programmation. J'estime que l'IA m'a fait gagner au moins un mois de temps de programmation au cours de l'année écoulée.

Mais le code que l'IA a aidé à écrire est mauvais. Et je passe du temps à corriger les cas que l'IA n'a pas anticipés, à faire de la rétro-ingénierie sur l'architecture de codage de l'IA, puis à remettre ce code en production. On est loin de la délégation des tâches de codage.

Qu'en pensez-vous ? L'IA remplacera-t-elle les programmeurs ou créera-t-elle une demande de compétences en programmation ? Êtes-vous déjà en train de vous perfectionner ? Pensez-vous que l'IA vous obligera à vous perfectionner ?

Source : "ZDNet.com"

annexe

Selon Gartner, l'IA générative nécessitera la mise à niveau des compétences de 80 % des ingénieurs d'ici 2027

STAMFORD, Connecticut, 3 octobre 2024

À plus long terme, les entreprises auront besoin d'ingénieurs en IA pour créer des applications optimisées par l'IA

D'ici 2027, [l'IA générative \(GenAI\)](#) engendrera de nouveaux rôles dans l'ingénierie logicielle et les opérations, nécessitant que 80 % des ingénieurs se perfectionnent, selon Gartner, Inc.

« Les affirmations audacieuses sur les capacités de [l'IA](#) ont conduit à spéculer sur le fait que l'IA pourrait réduire la demande d'ingénieurs humains, voire les supplanter entièrement », a déclaré [Philip Walsh](#), analyste principal chez Gartner. « Alors que l'IA va transformer le futur rôle des ingénieurs logiciels, l'expertise et la créativité humaines seront toujours essentielles pour fournir des logiciels complexes et innovants. »

Les analystes de Gartner s'attendent à ce que l'IA ait un impact sur le rôle de l'ingénierie logicielle de trois manières :

À court terme, l'IA fonctionnera dans des limites

- Les outils d'IA généreront des gains de productivité modestes en améliorant les modèles de travail et les tâches des développeurs existants. Les avantages de productivité de l'IA seront particulièrement significatifs pour les développeurs seniors des organisations ayant des pratiques d'ingénierie matures.

À moyen terme, l'émergence des agents IA repoussera les limites

- Les agents IA transformeront les modes de travail des développeurs en leur permettant d'automatiser entièrement et de décharger davantage de tâches. Cela marquera l'émergence de l'ingénierie logicielle native de l'IA, lorsque la plupart du code sera généré par l'IA plutôt que par des humains.

« À l'ère de l'IA native, les ingénieurs logiciels adopteront une mentalité de « priorité à l'IA », où ils se concentreront principalement sur l'orientation des agents IA vers le contexte et les contraintes les plus pertinents pour une tâche donnée », a déclaré Walsh. Cela rendra les compétences en ingénierie des invites en langage naturel et en génération augmentée de récupération (RAG) essentielles pour les ingénieurs logiciels.

À long terme, les progrès de l'IA repousseront les limites et marqueront l'essor de l'ingénierie de l'IA

- Alors que l'IA rendra l'ingénierie plus efficace, les organisations auront besoin d'ingénieurs logiciels encore plus qualifiés pour répondre à la demande croissante de logiciels basés sur l'IA.

« La création de logiciels basés sur l'IA exigera un nouveau type de professionnel du logiciel, l'ingénieur en IA », a déclaré Walsh. « L'ingénieur en IA possède une combinaison unique de compétences en ingénierie logicielle, en science des données et en IA/apprentissage automatique (ML), des compétences très recherchées. »

Selon une enquête Gartner menée au quatrième trimestre 2023 auprès de 300 organisations américaines et britanniques, 56 % des dirigeants en ingénierie logicielle ont estimé que l'ingénieur en IA/apprentissage automatique (ML) était le rôle le plus demandé pour 2024, et ils ont estimé que l'application de l'IA/ML aux applications était le plus grand déficit de compétences.

Pour soutenir les ingénieurs en IA, les entreprises devront investir dans des plateformes de développement d'IA. Ces plateformes aideront les entreprises à développer plus efficacement leurs capacités d'IA et à intégrer l'IA dans des solutions d'entreprise à grande échelle. « Cet investissement obligera les entreprises à perfectionner les compétences de leurs équipes d'ingénierie des données et des plateformes afin qu'elles adoptent des outils et des processus qui favorisent l'intégration et le développement continu des artefacts d'IA », a déclaré M. Walsh.

sur ces conférences pour apprendre comment amplifier l'impact de la technologie, des connaissances et des tendances qui façonnent l'avenir de l'informatique et des entreprises. Suivez les actualités et les mises à jour des conférences sur X en utilisant [#GartnerSYM](#) et sur la [salle de presse de Gartner](#) .