

Livrable individuel :
Qualité du logiciel et de la
maintenance

Contexte	2
L'impact sur la qualité de code du logiciel.....	2
Avantages sur la maintenance du logiciel.....	2
Conclusion.....	2

Contexte

Il y a de cela un an jour pour jour, j'utilisais l'IA en stage, en entreprise, sur la demande de mon tuteur pour le développement de leur logiciel. C'était à ce moment-là un outil de productivité et gage de vitesse puissant en recherche documentaire pour le code, un outil de débogage mais surtout une source génératrice de code pour l'amélioration de la performance d'une boucle ou l'implémentation d'une animation par exemple. Au jour d'aujourd'hui, cette IA est beaucoup plus puissante qu'on pouvait l'imaginer : elle est capable de générer une application de A à Z, fonctionnel à partir d'un simple prompt. Cette évolution à vue émerger une nouvelle tendance qui est le "Vibe Coding" qui consiste à ce que nous avons défini juste avant. Le problème avec cela c'est que l'utilisateur qui n'est pas expérimenté en programmation aura tendances à faire confiance à l'IA et faire abstraction du code car "cela semble fonctionner".¹ Cependant, quel sont les impacts sur la qualité du logiciel et sa maintenance ?

L'impact sur la qualité de code du logiciel

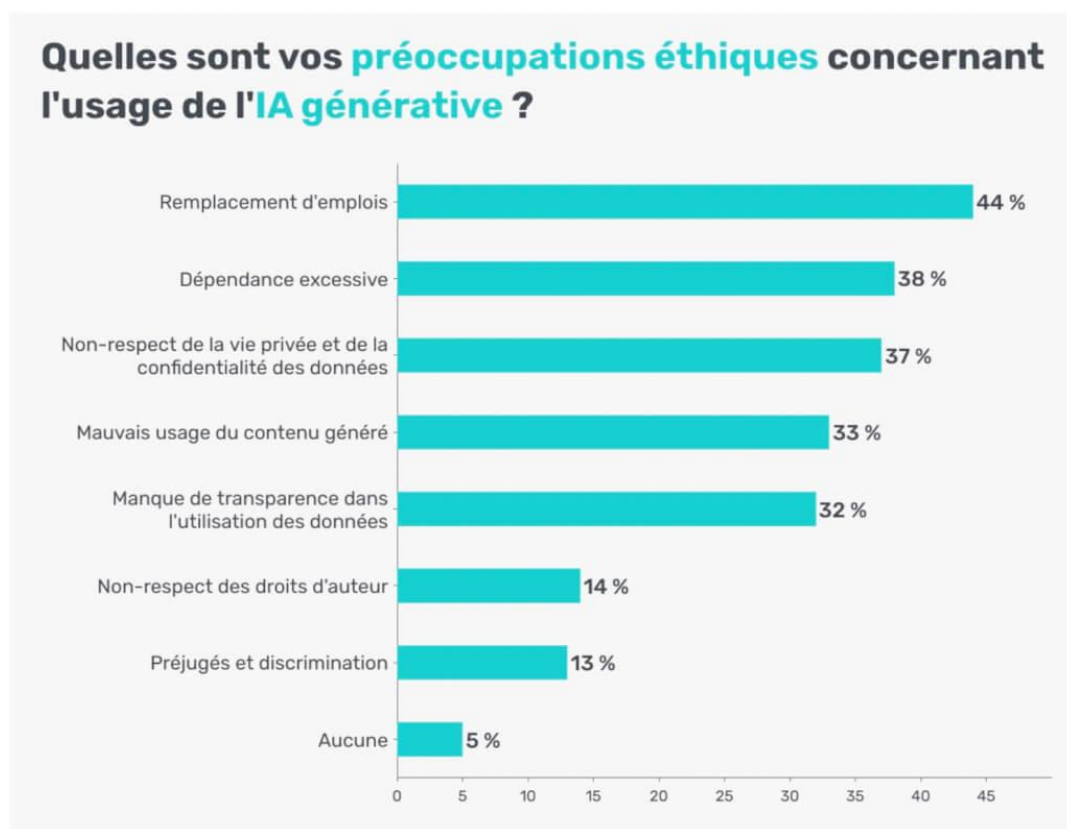
Comme nous l'avons introduit, l'IA peut désormais générer un logiciel entier qui fonctionne, mais malgré ça, elle n'a pas de conscience à proprement parlé et ne "comprend" pas le contexte et les exigences qui doivent être prises en compte lors du développement d'un logiciel. Cela s'explique par le fait que ses résultats sont toujours probabilistes et se basent des modèles courants qui existe déjà. Pour un logiciel innovant qui apporte un nouveau besoin et qui se veut démarquer des autres, par exemple, sans une intervention humaine et une revue du code pour vérifier qu'il correspond bien aux attentes, le logiciel sera finalement ce que l'on trouve généralement partout avec un code générique.

Le risque est que nous développeurs devenons trop dépendants de l'IA, ce qui nous conduira à nous reposer sur un code sans éthique. L'expertise est cruciale dans la programmation car un bon code est bien maintenu par la suite.² Ce qui

¹ Qu'est-ce que le vibe coding ? : <https://cloud.google.com/discover/what-is-vibe-coding>

² L'IA peut-elle améliorer la qualité du logiciel ? : <https://vaultinum.com/fr/blog/l-ia-peut-elle-ameliorer-la-qualite-du-logiciel>

nous amène à penser à la dimension social et juridique d'un code généré par IA, comme nous pouvons le voir sur ce graphique :



Graphique d'analyse de l'utilisation de l'IA générative en entreprise³

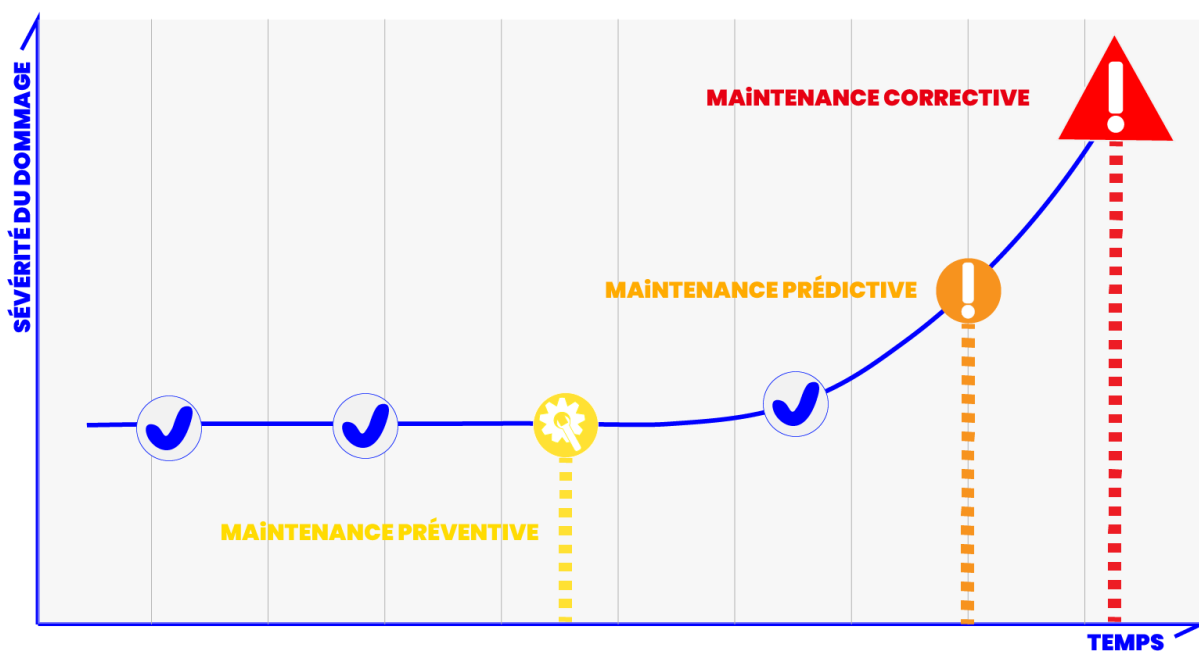
Le plus gros danger sont notamment les nouvelles vulnérabilités et failles liées à au développement avec l'IA comme le fait qu'ils suggèrent parfois des solutions de facilité, du code vulnérable (requêtes SQL non paramétrées menant à une injection SQL). En raison de cette confiance, il y a eu de cas où des copiés-collés de code proposés par chatgpt sans réelle compréhension, avait des failles incluses car il a incité à utiliser une librairie connue pour sa simplicité sans mentionner qu'elle était obsolète avec donc plein de failles dû à l'arrêt des mises à jour de sécurité.⁴

³ Les risques de l'IA de mieux en mieux cernés : <https://www.solutions-magazine.com/risques-ia-mieux-cernes/>

⁴ L'impact de l'IA sur les attaques, les failles et la sécurité logicielle : <https://dyma.fr/blog/limpact-de-lia-sur-les-attaques-les-failles-et-la-securite-logicielle/>

Avantages sur la maintenance du logiciel

Maintenant, penchons-nous sur les aspects positifs de l'IA dans le code avec la maintenance de celui-ci. La maintenance est une partie de la programmation assez chronophage lors de gros bugs, l'IA ici apporte une "maintenance préventive" qui contrairement à la maintenance classique de correction va permettre d'anticiper ces bugs et dysfonctionnement du logiciel. Cela se voit directement au niveau économique avec une réduction des coûts de maintenance de 12 à 18% (selon le Federal Energy Management Program (FEMP)) et niveau logiciel elle améliore la fiabilité de l'équipement de 35 à 45% et réduit les pannes de 70 à 75%.



Graphique des différentes phases de maintenance dans le temps⁵

Ce qu'apporte concrètement l'IA ici est une batterie de tests sur des milliers et millions de données dans plusieurs scénarios possible que nous développeur n'auront pas forcément imaginé. Comme sur des cas anomalies qui arrivent

⁵ La Maintenance Prédictive : un Enjeu Majeur pour l'Industrie : <https://bluedocker.com/maintenance-predictive-enjeu-industrie/>

souvent en productions que nous ne voyons pas en développement (nombres conséquents d'utilisateurs, requêtes massives, etc).

Conclusion

Dans le futur, l'intégration dans le processus de développement avec les IA est vouée à changer avec une capacité à comprendre le contexte et remédier aux problèmes actuels. Malgré qu'elle offre une rapidité, à ce jour c'est à nous développeur d'être responsable de la qualité du code qu'on produit et c'est pour cela que notre expertise et notre formation compte : L'IA ne nous remplacera pas, elle est un outil et un compagnon de programmation comme les écrivains dans leur processus de recherche documentaire par exemple.