

SAÉ – II. Livrable individuel



Analyse d'un risque lié aux IA génératives

sujet : réputation et appropriation du produit par le public

**Hichem
Zenaini
MAKEMAKE**

Phase 3 : Réputation et appropriation du code

Dans ce projet, on a d'abord codé à la main puis nous avons utilisé des IA pour voir la différence. On a pu voir que l'IA avait des algorithmes plus poussés et qu'elle le faisait rapidement (quelques secondes), mais le problème était qu'on ne comprenait pas le code. Du coup, je me suis posé la question : est-ce qu'un client va toujours nous faire confiance si on ne comprend pas ce qu'on code et qu'on ne sait pas expliquer, même si ça fonctionne ? Si le code plante et que l'IA n'arrive pas à le régler, comment faire ? Dans cette analyse, je vais expliquer pourquoi l'utilisation de l'IA peut être un risque pour l'image d'une entreprise et pourquoi c'est dur de s'approprier un code qu'on n'a pas écrit nous-mêmes.

En phase 1, on a tout fait à la main. C'était long et fatigant, surtout pour les calculs de distances, mais on savait exactement comment le programme marchait. En phase 2, avec l'IA, c'était l'inverse. Le code était là tout de suite, il marchait, mais je ne comprenais pas la logique derrière. Par exemple, l'IA a généré une fonction très courte pour optimiser les cellules de Voronoï. C'était efficace, mais quand on demande d'expliquer, on ne sera pas répondre. C'est là que j'ai compris le souci : si on ne peut pas expliquer notre code entre nous on ne pourra jamais le justifier devant un client. On devient un peu des personnes qui font du copier coller au lieu d'être des développeurs. Comme l'explique un article du Monde Informatique, le risque principal est que les développeurs perdent leurs compétences techniques à force de trop déléguer à l'IA.

Si on vend un logiciel à un client et qu'il y a un bug, on doit être capable de le réparer vite. Pendant le projet, on a eu un bug avec des points alignés et on a mis des heures à trouver le problème parce que le code de l'IA était trop complexe et différent de notre façon de faire. Si ça arrive en production, le client va se dire qu'on n'est pas sérieux et cela donnera une mauvaise image. Le client paye pour notre expertise pas pour qu'on demande à une IA de corriger les erreurs. Si ils se rendent compte qu'on utilise l'IA pour tout ils se diront qu'ils peuvent le faire eux-mêmes et on perdra toute crédibilité. C'est risqué pour notre image si on passe pour des gens qui copient-collent sans comprendre, on perd toute notre crédibilité. De plus la CNIL rappelle d'ailleurs que l'entreprise reste la seule responsable des outils numériques qu'elle utilise. Si l'IA fait une erreur, c'est nous qui devons en assumer les conséquences juridiques et assumer face au client.

Le piège c'est que l'IA nous donne un code qui semble être correcte alors qu'en fait il y a des bugs cachés. On teste avec 5 ou 10 points, ça marche et on est contents. Mais en testant plus sérieusement avec 100 points très serrés, on a vu que ça pouvait planter. Des experts sur Numérama ont d'ailleurs souligné que l'IA génère souvent du code qui a l'air propre mais qui cache des failles ou des erreurs de logique importantes. Si on livre un produit avec un bug caché qui n'apparaît qu'au bout de trois mois chez le client c'est une catastrophe pour notre image de marque. On risque de perdre des clients et de se faire une mauvaise réputation.

Conclusion

Pour finir, je pense que l'IA est un bon outil pour les petites tâches répétitives, mais on ne doit pas lui laisser faire tout le travail de réflexion. Il faut qu'on reste les maîtres de l'architecture de nos applications. on peut l'utiliser en temps que copilote demander des explications, comment faire ça et il faut comprendre ce qu'on implémente

Au début, je pensais que l'IA nous ferait gagner un temps, mais entre le temps passé à comprendre le code généré et celui à corriger des bugs, le gain n'est pas si grand que ça. Pour rester crédibles et garder une bonne réputation, on doit absolument continuer à comprendre chaque ligne de code. C'est l'humain qui doit garder la main sur la technique pour que le client reste en confiance. De plus l'IA crée des failles de sécurité

Sources :

- Le Monde Informatique : "IA générative et développement : attention à la perte de compétences des équipes."
- Numérama : "Pourquoi le code généré par IA pose des problèmes de sécurité et de maintenance."
- CNIL : "Guide sur l'usage des outils d'IA : maîtriser les risques et la responsabilité en entreprise."