

Résultats et constats de la phase 2

Expérimentation de l'outil d'intelligence artificielle générative
GitHub Copilot au ministère de la Cybersécurité et du Numérique

2025

Éléments de contexte

L'humain au cœur de l'expérimentation

Phase 1 : Recommandation de poursuivre l'expérimentation avec une phase 2 afin d'approfondir les connaissances acquises grâce à une approche renouvelée;

Phase 2 : Réalisée par le CQEN et inscrite dans la démarche gouvernementale d'expérimentation coordonnée par le SMADPIG

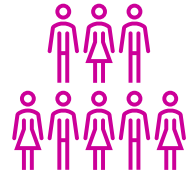
- Participation unique du MCN, en partenariat avec Microsoft;
- Autonomie sur les objectifs et les analyses;
- Résultats nourrissent à la fois les réflexions ministérielles et gouvernementales.

L'expérimentation en chiffres



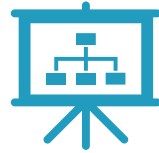
3 1/2

mois



30

personnes
participantes



6

équipes
suivies



3h

de formation
obligatoire



3

sondages

3

rencontres du
comité directeur

4

personnes
ambadrices

34

cas recueillis

3

ateliers
de partage

25

personnes
interviewées

Méthodologie

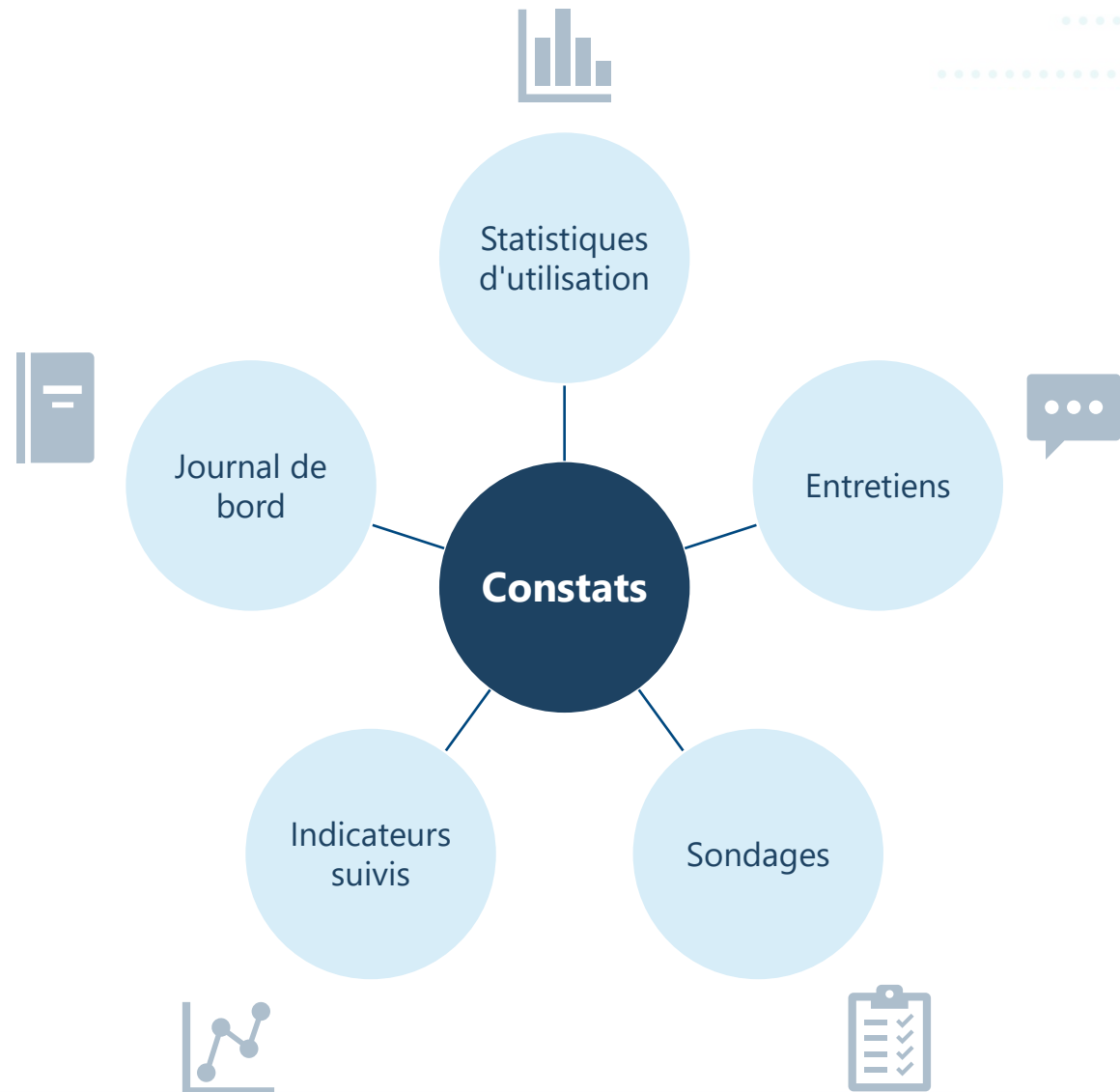
Durée : du 27 mars au 9 juillet 2025

30 participants

6 équipes dont les activités concernent principalement le développement

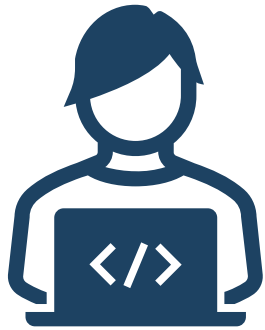
30 cas d'utilisation soumis via le journal de bord

4 cas d'usage suivis grâce à des mesures avec et sans l'outil



Un profil assez uniforme chez les participants

Utilisation déjà fréquente de l'IA générative par un groupe expérimenté



- 70 % ont **10 ans et plus d'expérience**
- 73 % avaient **utilisé préalablement l'IA** et pour la plupart à une **fréquence journalière ou hebdomadaire**
- 62% sont **développeurs *full stack***
- 96% **percevaient de façon positive** l'introduction de l'outil dans leur environnement de travail.



Quels sont les effets de l'utilisation de GitHub Copilot sur...



la productivité
des développeurs?

1 constat



la qualité
du code produit?

3 constats



le bien-être
des développeurs?

2 constats

4 cas d'usage suivis



Constat et résultats

En matière de productivité



1 L'outil est un puissant accélérateur de productivité qui permet aux développeurs de consacrer plus de temps à la résolution de problèmes complexes.

- Les tâches répétitives sont optimisées.
- Le recours aux recherches traditionnelles, une approche chronophage, est diminué.
- 92 % estiment que l'utilisation de l'outil permet de gagner du temps.
- 80 % considèrent que perdre l'accès à l'outil aurait un impact négatif sur leur productivité, allant de modéré (52 %) à important (28 %).



Avant, je pouvais passer plusieurs heures à chercher des solutions sur Google. Aujourd'hui, j'obtiens des réponses quasi instantanées grâce à Copilot, ce qui me permet de rapidement trouver une piste de solution.



Constats et résultats

En matière de qualité



2

La qualité du code est rehaussée, mais dans une moindre mesure que le gain en productivité.

- 52 % observeraient une diminution de la qualité de leur code s'ils ne pouvaient plus utiliser l'outil, contre 80 % qui anticiperaient une baisse de leur productivité.
- L'outil aide à détecter des optimisations difficiles à repérer à l'œil nu et à adopter de meilleures pratiques.
- La vigilance reste nécessaire, car certaines suggestions peuvent introduire du code inutile ou redondant.

Copilot m'aide à faire des revues de code plus précises et approfondies, en pointant des améliorations ou des incohérences que je n'aurais peut-être pas remarquées seul.



3

Les propositions générées par l'outil font l'objet d'une évaluation critique continue.

- La confiance en l'outil reste partielle : 44 % lui font plutôt confiance, tandis que 48 % lui font moyennement confiance.
- Toutes les personnes disent réviser ou modifier le code proposé par l'outil avant de l'utiliser.
- Le risque principal relevé réside dans une appropriation non critique de l'outil, en particulier dans les environnements où les pratiques de validation ne sont pas solidement établies.



C'est rare que je le copie, puis que je le colle directement. [...] on tricote autour.



Le risque [...] c'est qu'ils [les juniors] acceptent aveuglément tout ce que l'intelligence artificielle suggère.



4

La révision de code pourrait être facilitée par l'utilisation de l'outil.

- En amont de la révision, l'outil aide les développeurs à repérer les erreurs et à s'assurer du respect des normes, allégeant ainsi la tâche des réviseurs.
- **Mais**, l'utilisation de l'IA par les développeurs pose le risque d'une surcharge de contenu généré à réviser (quantités de lignes générées et documentation abondante).
- Les réviseurs peuvent eux-mêmes utiliser l'outil pour proposer des recommandations plus poussées, détecter en avance des blocages et uniformiser le code plus facilement.

Le rôle va changer, mais la barrière humaine doit toujours rester

La force de l'outil, c'est d'aider à réfléchir.

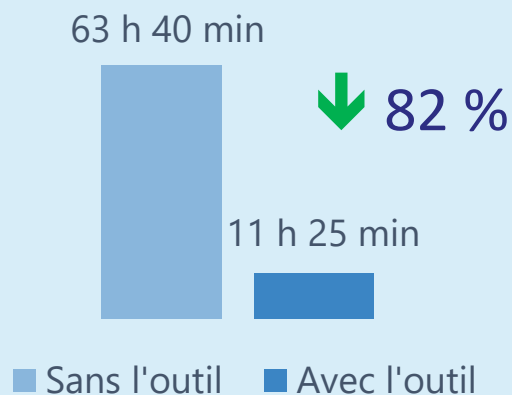


Appréciation de l'outil selon les cas d'utilisation

En termes de productivité et de qualité

Des cas d'usage mesurés démontrant des gains sans équivoque en termes de productivité

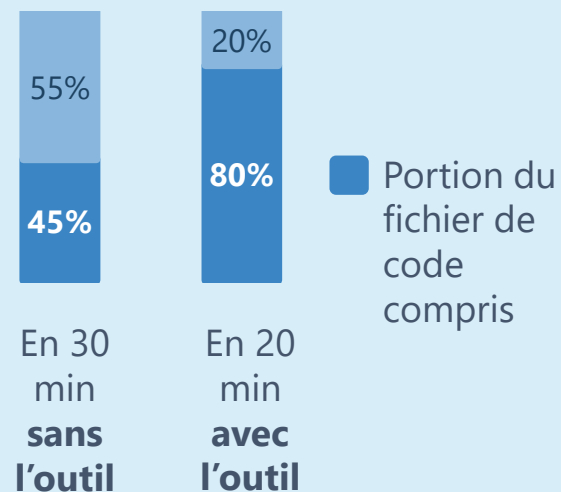
Conversion de langage pour 8 Notebook



Création de test unitaires



Acquisition de connaissances



Documentation via le développement de 3 pages Web



Complétion, génération et transformation de code

Cas d'utilisation récoltés dans les journaux de bord

Des gains significatifs, allant jusqu'à une économie de 20 jours, ont été observés.

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	4	5	0	9
	Inchangée	1	0	0	1
	Diminuée	0	0	0	0
Total		5	5	0	10

Cas d'usage mesuré - Hors journal de bord

Dans un service, le langage de huit *Notebooks* devait être converti. Les personnes participantes de cette équipe ont donc mesuré le temps requis pour mener cet exercice avec et sans l'outil et il est ressorti que ce travail qui aurait pris deux semaines (63 heures et 40 minutes) sans l'IA générative a été fait en moins de deux jours (11 heures et 25 minutes) avec l'outil. **L'utilisation de l'outil a permis de réduire le temps de travail d'environ 82 %.**

Refactorisation et optimisation du code

Cas d'utilisation récoltés dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	4	0	0	4
	Inchangée	0	0	0	0
	Diminuée	0	0	0	0
Total		4	0	0	4

De quelques minutes à cinq ou six jours ont été sauvés avec l'utilisation de l'outil d'IA générative GitHub Copilot.



C'était un peu plus efficace et plus intelligent qu'un simple « search and replace », d'où un gain. Les comportements un peu erratiques de la navigation/suggestion avec "tab" n'ont pas permis d'avoir autant de gains qu'espéré.

Dans les dernières 6 semaines sur cette tâche, j'ai économisé, selon mes estimés, 5.5 jours de travail, tant à moi qu'à d'autres collègues sur lesquels j'aurais dû dépendre en les appelant pour des questions. Des réponses en quelques secondes plutôt que des heures de recherche.

Génération de tests

Cas d'utilisation récoltés dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	1	2	0	3
	Inchangée	0	0	0	
	Diminuée	0	1	0	1
Total		1	2	1	4

Des pertes de 5 à 10 minutes à des gains pouvant aller jusqu'à 3 heures ont été rapportés.

Cas d'usage mesuré - Hors journal de bord

Dans une direction, des mesures ont été effectuées lors de création de tests unitaires. Lors de la première mesure, le test unitaire a été créé en une heure sans l'outil et en 36 minutes avec celui-ci. Pour les deux mesures subséquentes, le même résultat a été obtenu : trois heures ont été nécessaires pour créer chaque test sans l'outil et le temps requis a diminué à une heure par test avec l'outil.

L'utilisation de l'outil a permis de réduire le temps de travail d'environ 63 %.

Correction et débogage

Cas d'utilisation récoltés dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	2	0	0	2
	Inchangée	0	0	0	0
	Diminuée	0	0	1	1
Total		2	0	1	3

Des gains de 5 minutes et de plusieurs heures ont été estimés dans les 2 cas. L'autre cas présente une perte de 5 minutes, qui aurait pu être plus importante si le développeur n'avait pas eu la présence d'esprit de mettre en doute l'outil d'IA générative GitHub Copilot.



Ne connaissant pas la fonctionnalité qu'il a finalement suggéré, je me dirigeais vers un contournement (« hack »).

Lors d'un échange avec un participant, il a été mentionné que l'utilisation de l'outil d'IA générative GitHub Copilot est un atout pour comprendre pourquoi le code ne fonctionne pas.

Acquisition de connaissances

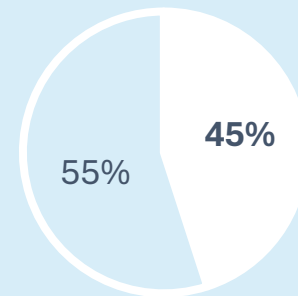
Cas d'utilisation récoltés dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	2	0	0	2
	Inchangée	0	0	0	0
	Diminuée	0	0	0	0
Total		2	0	0	2

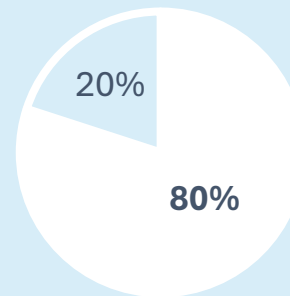
L'utilisation de l'outil a permis d'être de 3 à 4 fois plus rapide dans les 2 cas soumis.

Cas d'usage mesuré - Hors journal de bord

Un test a été mené pour déterminer l'apport de l'outil dans la compréhension d'un fichier de code :



En 30 minutes
sans l'outil



En 20 minutes
avec l'outil

■ Portion de code compris

Documentation et génération de commentaires

Cas d'utilisation récolté dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	1	0	0	1
	Inchangée	0	0	0	0
	Diminuée	0	0	0	0
Total		1	0	0	1

Une vélocité 4 fois supérieure a été observée.

Cas d'usage mesuré - Hors journal de bord

Un développeur a comparé le développement de trois pages de documentation avec et sans l'outil. Il est ressorti que quatre heures ont été nécessaires pour faire les trois pages sans l'outil d'IA générative GitHub Copilot comparativement à moins de 35 minutes avec celui-ci. révision accordée. **L'utilisation de l'outil a permis de réduire le temps de travail d'environ 85 %.**

Intégration d'interface d'intégration d'application (API)

Cas d'utilisation récoltés dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	1	0	0	1
	Inchangée	0	0	0	0
	Diminuée	0	1	0	1
Total		1	0	1	2

L'utilisation de l'outil d'IA générative GitHub Copilot a permis de sauver cinq heures de travail et la qualité s'en est trouvée meilleure, surtout dans la gestion des codes d'erreur.

Le 2^e cas concerne l'activation et l'intégration d'une technologie externe dans le code source d'un logiciel du MCN. Plutôt que de proposer une réponse au besoin exprimée, l'outil a reproduit le contenu déjà rédigé par la personne développeuse. Une perte de 15 minutes a été constatée.

Révision de code

Cas d'usage récolté dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	0	0	0	0
	Inchangée	0	1	0	1
	Diminuée	0	0	0	0
Total		0	1	0	1

Aucun gain de temps n'a été obtenu, mais l'utilisation de l'outil d'IA générative GitHub Copilot a généré une conclusion sur la demande de tirage presque identique à ce qui aurait été fait par un humain. Cette conclusion a permis de confirmer les points de la personne réviseuse lors de cette activité. À noter que, parfois, l'outil suggère d'autres éléments, qui ne sont pas toujours pertinents.

Gestion des branches

Cas d'utilisation récolté dans les journaux de bord

		Qualité			Total
		Augmentée	Inchangée	Diminuée	
Productivité	Augmentée	0	1	0	1
	Inchangée	0	0	0	0
	Diminuée	0	0	0	0
Total		0	1	0	1

Une seule personne développeuse a soumis un cas d'utilisation sur la comparaison d'une branche actuelle de son code avec la branche principale. Cette comparaison a permis d'appliquer les changements dans une autre branche à la suite à d'une refactorisation majeure du code par un coéquipier. La personne développeuse a ainsi gagné 50 % du temps pour effectuer la tâche en se fiant à l'analyse de l'outil d'IA générative GitHub Copilot dans la recherche de code et ignorer les fichiers où il n'y avait que des correctifs de style (« linting »).



Constats et résultats

En matière de bien-être



5

L'outil réduit la charge mentale des personnes développeuses.

- L'automatisation et la simplification de tâches répétitives et chronophages permet de de réduire la charge mentale et le stress.
- La disponibilité immédiate de solutions ou de pistes d'approche rend plus confiant et diminue le stress lié à la performance.



Travailler sur des tâches à plus grande valeur ajoutée nécessitant plus de réflexion et déléguer en grande partie les plus simples et ennuyantes vaut de l'or, et Copilot me le permet.



6 L'outil facilite l'apprentissage, mais peut donner une fausse impression de maîtrise.

- L'outil agit comme un mentor constamment accessible qui facilite l'apprentissage et renforce la confiance des personnes développeuses lors de l'exécution de tâches moins familières.
- 80 % estiment que l'outil les a aidés à développer de nouvelles compétences ou à comprendre de nouveaux concepts.
- Quelques personnes reconnaissent cependant qu'elles ne comprennent pas toujours en profondeur ce qu'elles produisent à l'aide de l'outil, même si le résultat est fonctionnel.



C'est comme si j'avais un expert toujours présent avec moi pour répondre à des questions complexes.



Même si on ne comprend pas tout, on arrive à un résultat probant.

Quels sont les effets de l'utilisation de GitHub Copilot sur...



la productivité
des développeurs?

L'outil est un puissant accélérateur de productivité dans la majorité des cas.



la qualité
du code produit?

Le code généré n'est pas exempt de défauts et doit être évalué avec une rigueur constante.



le bien-être
des développeurs?

En allégeant la charge mentale et le stress, Copilot libère du temps pour des tâches stimulantes.



Constats et résultats

En matière d'adoption et d'utilisation



7 **GitHub Copilot s'impose comme un allié incontournable: entre enthousiasme, dépendance et inquiétude de devoir s'en passer**

- Entre la mi-parcours et la fin de l'expérimentation, le taux de participants utilisant l'outil plusieurs fois par jour passe de 39 % à 60 %.
- Intégration forte dans les pratiques de travail
- Essentiel pour la productivité de plusieurs personnes
- Génère à la fois motivation et appréhension



Je n'imagine pas ma journée de travail sans Copilot.

8 **GitHub Copilot n'est pas seulement un outil, il est perçu comme un collègue numérique, parfois même comme un mentor.**

- Influence perçue au-delà des performances techniques
- Présence rassurante et appui constant
- Personnification fréquente dans les témoignages
- Dimension relationnelle et émotionnelle ajoutée à son usage
- Redéfinit le rapport des professionnels face à l'IA



Ce qui me manquerait le plus? Un ami.



9 Une forte satisfaction traduit une réelle ouverture à l'IA générative, mais conditionnelle à lever les irritants techniques

- Majorité des participants exprime une satisfaction élevée
- Prédilection favorable à l'appropriation et à l'intégration de l'IA générative
- Terrain propice pour un déploiement à plus grande échelle
- Importance de tenir compte des nuances et contextes d'usage
- Adoption conditionnelle à la résolution des irritants et des limites

“Ça accélère tellement le développement que ce serait comme interdire Google ou Stack Overflow

“Sans Copilot, ça va être dur [...] c'est de la drogue.

10 Même imparfait, GitHub Copilot est jugé préférable à l'absence d'outil d'assistance.

- Sous performance de l'outil s'explique:
 - 68% des rejets sont dus à des suggestions non pertinentes
 - Limites liées aux attentes complexes ou trop précises
 - Besoin d'améliorer les compétences en rédaction de requêtes (prompt engineering)
- Reconnu comme une avancée importante pour le développement au MCN

“ Il y a des pièges à éviter et de meilleurs modèles de langage que ceux testés, mais ce serait déjà un grand pas en avant que de pouvoir l'utiliser.



11

Même en cas de retrait, la majorité poursuivraient l'utilisation de l'IA générative en se tournant vers des solutions alternatives.

- La plupart s'attendent à une perte d'efficacité en cas de retrait de l'outil.
- Des solutions alternatives seraient envisagées, même si elles sont moins pratiques (étant non intégrées dans l'environnement de développement) ou moins sécuritaires.

Comportement envisagé	
J'utilise déjà un ou plusieurs outils d'IA générative en complément de GitHub Copilot et je continuerais de les utiliser.	32 %
Je chercherais activement un outil qui pourrait remplacer GitHub Copilot.	40 %
Je testerais quelques outils pour voir s'ils répondent à mes besoins.	16 %
Je n'utiliserais pas d'outil d'IA générative tant qu'on ne m'en propose pas un autre.	12 %
Je n'ai pas l'intention de continuer à utiliser d'outil d'IA générative dans mon travail de développement.	-

Profils et contexte à privilégier pour l'attribution de l'outil



Profils dont la majorité de la charge de travail concerne :

- Des tâches de développement applicatif
- Des langages soutenus par l'outil
– lien vers l'information
- Des activités qui ne requièrent pas des privilèges élevés en termes de sécurité.



Éléments de **contexte** requis :

- Une utilisation d'outils dans lequel GitHub Copilot peut être intégré
- Une équipe dont le processus de révision est bien établi et fait de façon régulière
- Une ou des ressources de soutien prêtes à intervenir rapidement en cas de problématique technique

Actions clés en cas de déploiement



**Élaborer une stratégie
d'implantation et ancrer
l'outil dans le discours
organisationnel**



**Soutenir et suivre
l'adoption et ajuster en
continu**



**Renforcer la sécurité et la
gouvernance**



**Favoriser l'équité
et l'inclusion**



**Fournir de la formation et
l'accompagnement
différencié**



**Prévoir une stratégie
de continuité**

Hypothèses à explorer

D'autres solutions ou outils d'IA générative pourraient être considérés



Installation locale d'un modèle d'assistant de programmation

Permet de garder une indépendance vis-à-vis un service externe, d'avoir une meilleure protection des données et de garder une autonomie opérationnelle comme la solution fonctionne sans licence tierce.



Solutions alternatives

Des alternatives crédibles comme *Tabnine*, *Codeium* ou d'autres solutions identifiées par Gartner redistribuent les cartes du marché. Ces solutions offrent souvent des approches différenciées.

En ayant en tête les valeurs et priorités du MCN, ces solutions pourraient être étudiées afin d'évaluer le potentiel de ces alternatives à court et à moyen termes.