

# Dossier de Veille Technologique : L'IA et la Programmation

Période couverte : septembre – octobre 2024

Thème : L'IA et la programmation

## Table des matières

Dossier de Veille Technologique : L'IA et la Programmation.....	1
1. Introduction.....	2
2. Présentation du sujet.....	2
2.1 Qu'est-ce que l'IA en programmation ?.....	2
2.2 Motivation du choix.....	2
3. Outils utilisés et Méthodes de recherche.....	2
3.1 Méthodes Pull et Push.....	2
Méthode Pull :.....	2
Méthode Push :.....	2
3.2 Outils utilisés pour la veille.....	3
4. L'IA et la programmation : Usages, avantages et inconvénients.....	3
4.1 Usages actuels.....	3
4.2 Avantages.....	3
4.3 Inconvénients.....	3
5. Exemples d'entreprises et projets utilisant l'IA dans la programmation.....	4
6. Dates importantes et évolution future de l'IA en programmation.....	4
7. Sources.....	5
7.1 Validation des sources.....	5
8. Conclusion.....	5
9. Annexes : Copies d'écran des outils de Push.....	6



# 1. Introduction

Dans ce dossier de veille technologique, nous nous intéressons à l'impact de l'intelligence artificielle (IA) sur la programmation. L'IA représente actuellement une transformation majeure dans de nombreux secteurs, dont celui du développement logiciel.

J'ai choisi d'analyser comment l'IA influence la programmation, en observant les outils actuels, leurs avantages, inconvénients, ainsi que les perspectives d'avenir de cette technologie. Le sujet a été étudié entre septembre et octobre 2024.

## 2. Présentation du sujet

### 2.1 Qu'est-ce que l'IA en programmation ?

L'intelligence artificielle (IA) en programmation désigne l'utilisation de technologies basées sur des LLM (Large Language Models) pour assister les développeurs. Ces outils peuvent générer du code, détecter des erreurs et automatiser des tâches répétitives dans le processus de développement.

### 2.2 Motivation du choix

Le choix de ce sujet repose sur l'impact grandissant de l'IA dans le monde de la programmation. Aujourd'hui, de nombreux outils comme **GitHub Copilot** ou **ChatGPT o1** permettent aux développeurs de générer des blocs de code ou des solutions entières, réduisant ainsi le temps de développement et augmentant la productivité. Le sujet est particulièrement pertinent car il modifie fondamentalement la façon dont les logiciels sont créés.

A titre personnel, je suis convaincu qu'il est important de s'adapter à cette nouvelle façon de développer pour rester compétitif.

## 3. Outils utilisés et Méthodes de recherche

### 3.1 Méthodes Pull et Push

#### Méthode Pull :

Le pull consiste à rechercher manuellement des informations. J'ai utilisé :

- **Youtube** où il est possible de trouver de nombreuses personnes donnant leurs avis. Par exemple, il est possible de trouver sur la chaîne Hardisk TV des rediffusions d'émissions où trois technophiles discutent de sujets liés à l'actualité technologique. Leurs expériences dans divers domaines rendent leurs avis pertinents.

#### Méthode Push :

Le push consiste à recevoir des informations de manière automatisée via des abonnements à des outils. J'ai utilisé :

- **Google Alerts** pour recevoir des listes d'articles lors de leurs parutions en se basant sur des mots clés.

- **Newsletters** indépendantes agrégeant de nombreuses informations autour de l'Intelligence Artificielle en général.

Des captures d'écran de ces outils sont disponibles en annexe.

### 3.2 Outils utilisés pour la veille

Lors de ma veille, j'ai donc utilisé les outils suivants :

- **Google Alerts**
- **Des newsletters**
- **Youtube**

## 4. L'IA et la programmation : Usages, avantages et inconvénients

### 4.1 Usages actuels

Comme expliqué dans l'article [L'IA Générative va-t-elle mettre en péril l'approche low code no code](#), l'IA générative ne remplace pas les approches du code déjà existantes. Elle est un outil venant assister un développeur. Dans l'extrait [Ces IA savent programmer !](#) L'approche où l'IA vient simplement assister est très bien décrite. Mon oncle, développeur indépendant, a comparé le fait d'avoir un assistant IA à celui « d'avoir quelqu'un au-dessus de son épaule ».

L'IA est de plus en plus utilisée dans la programmation avec des outils tels que :

- **GitHub Copilot** : Un assistant de code alimenté par GPT qui aide à écrire du code plus rapidement.
- **DeepCode** : Analyse le code pour identifier des erreurs ou des optimisations possibles.
- **Replit Ghostwriter** : Fournit des suggestions de code en temps réel.

### 4.2 Avantages

Dans l'article [Du « boulier » à la « tronçonneuse » : comment l'IA va continuer à transformer les équipes de développement](#), l'arrivée de l'IA est vue comme une cohabitation plus qu'un remplacement. Il serait alors nécessaire de repenser l'organisation des équipes et de former les développeurs à l'utilisation de l'IA.

- **Gain de temps** : L'IA automatise des tâches répétitives et accélère l'écriture de code.
- **Réduction des erreurs** : Les systèmes d'IA peuvent détecter les bugs avant qu'ils n'aient un impact.
- **Productivité accrue** : Les développeurs peuvent se concentrer sur des tâches plus complexes tandis que l'IA gère les détails.

### 4.3 Inconvénients

Il faut noter que tout le monde n'est pas d'accord sur l'ampleur du gain de temps apporté par l'IA. L'article [IA générative dans la programmation : des bénéfices surestimés ?](#) Pointe du doigt le temps perdu à vérifier le code généré.

Il y a aussi des inquiétudes quant à la sécurité du code généré par IA. L'article [les Copilot et les applications low-code sont un cauchemar pour la sécurité informatique](#) décrit le problème de « coder sans rien y connaître ». C'est dans ce contexte que l'ANSSI et le BSI ont publié des [bonnes pratiques de sécurité sur la génération de code par l'IA](#).

Un autre inconvénient majeur de la programmation assistée par IA est la maintenance. En effet, il est très difficile de maintenir un code que l'on a pas écrit. Cela est illustré par l'article [Les assistants IA au développement inventent des noms de paquet](#).

## 5. Exemples d'entreprises et projets utilisant l'IA dans la programmation

L'entreprise liée au sujet de ce rapport ayant fait le plus parler d'elle ces derniers temps est Poolside. Dans son article [Poolside lève 500 millions de dollars pour son assistant au développement logiciel](#), l'usine digitale parle de l'arrivée d'un nouvel acteur dans les solutions de développement assisté par IA. Une entreprise n'ayant pas encore de produit commercialisé a réussi à récolter plusieurs centaines de millions d'euros pour développer son produit. Une telle confiance des investisseurs laisse envisager de belles perspectives d'avenir pour le secteur des IDE avec assistant IA.

Voici d'autres entreprises utilisant des IA pour programmer :

- **GitHub (Microsoft)** : Utilise l'IA dans **GitHub Copilot** pour aider les développeurs à générer des segments de code en temps réel.
- **Replit** : Propose des outils de développement assistés par l'IA, permettant à des développeurs de tout niveau de bénéficier de suggestions intelligentes.

## 6. Dates importantes et évolution future de l'IA en programmation

Voici les dates importantes de l'évolution du développement assisté par IA :

- **2020** : Lancement de **GPT-3**, qui a démocratisé l'utilisation de LLM.
- **2021** : Introduction de **GitHub Copilot** qui a intégré CoPilot dans des solutions dédiées aux développeurs.

L'utilisation de l'IA en tant qu'assistant est plus pertinente via des modèles de LLM intégrés aux IDE, où l'IA a correctement accès à la totalité du code concerné.

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."

## 7. Sources

### 7.1 Validation des sources

- **L'IA Générative va-t-elle mettre en péril l'approche low code no code**  
J'ai validé cette source car elle est récente, pertinente et rédigée par un expert du développement, avec des exemples concrets sur l'impact de l'IA sur les outils low-code.
- **Ces IA savent programmer !**  
J'ai validé cette source car les personnes s'exprimant, bien qu'ils ne soient pas tous experts, sont des utilisateurs de Cursor et ont généralement des avis pertinents sur les autres sujets.
- **Du « boulier » à la « tronçonneuse » : comment l'IA va continuer à transformer les équipes de développement**  
J'ai validé cette source car elle est rédigée par un expert reconnu et offre des exemples concrets de l'impact de l'IA sur les équipes de développement, avec des informations récentes.
- **IA générative dans la programmation : des bénéfices surestimés ?**  
J'ai validé cette source car elle propose une analyse critique des limites des IA en programmation, avec des exemples d'outils actuels et une perspective actualisée.
- **les Copilot et les applications low-code sont un cauchemar pour la sécurité informatique**  
J'ai validé cette source car elle traite des enjeux de sécurité liés aux IA en développement, un sujet crucial et récemment discuté par des experts en cybersécurité.
- **Les assistants IA au développement inventent des noms de paquet**  
J'ai validé cette source car elle aborde un problème technique concret rencontré avec les IA, étayé par des exemples récents provenant de développeurs.
- **Poolside lève 500 millions de dollars pour son assistant au développement logiciel**  
J'ai validé cette source car elle provient d'un média financier fiable et traite d'une actualité récente sur l'investissement dans les assistants IA pour le développement.

## 8. Conclusion

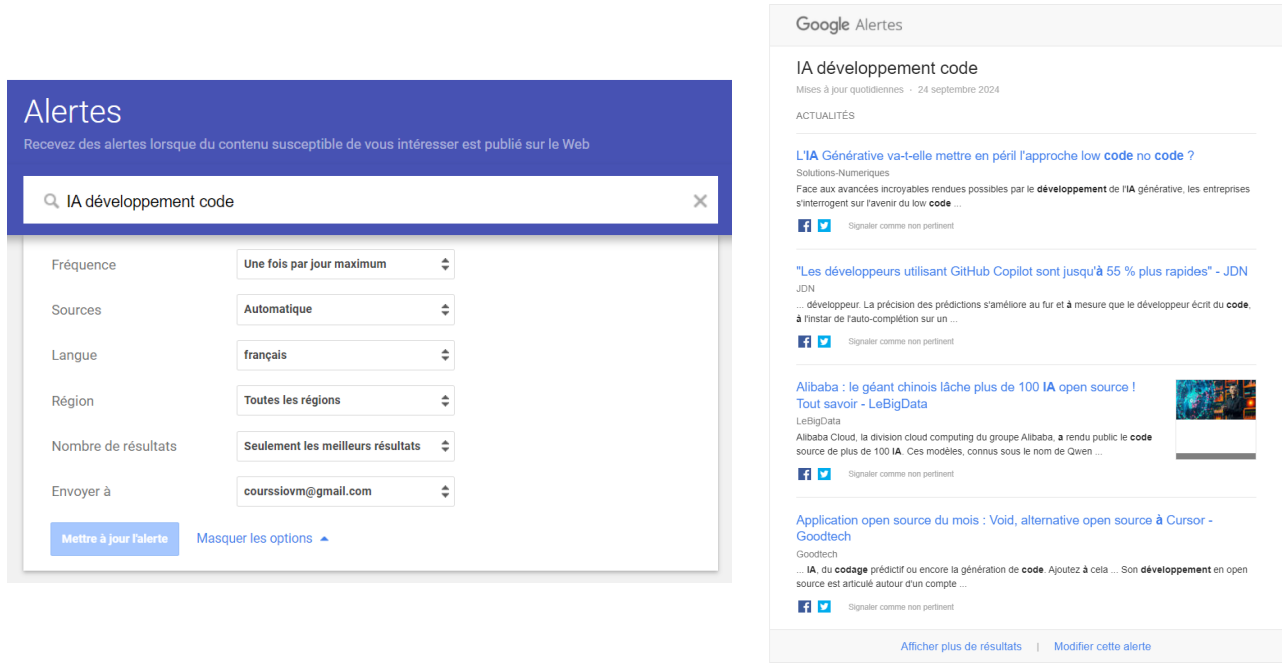
L'intelligence artificielle a le potentiel de transformer le domaine de la programmation en rendant certaines tâches plus rapides et plus efficaces. Cependant, à l'heure actuelle, l'IA n'est capable de générer que des morceaux de code et il est nécessaire d'avoir des compétences pour vérifier la viabilité du dit code.

Les outils comme **GitHub Copilot**, **DeepCode** ou encore **Cursor** représentent déjà des révolutions, et l'avenir nous réserve probablement des innovations encore plus surprenantes. Suivre cette évolution à travers la veille technologique est essentiel pour les développeurs souhaitant rester à jour et compétitif.

# 9. Annexes : Copies d'écran des outils de Push

Tous les articles cités dans ce rapport sont disponibles dans le dossier dédié à la veille sur NAS et sur mon Google Drive.

Capture d'écran de la configuration et d'un rapport de l'outil Google Alerts :



Voici un exemple de newsletter que j'ai pu recevoir lors de ma veille :

**AI News**

**OPENAI**  
**OpenAI's New Model: The Future of AI Reasoning?**

OpenAI has recently unveiled the **o1 model series**, marking a significant advancement in AI's ability to think and reason. The o1-preview and o1-mini models are now accessible to all ChatGPT plus/team users.

**Details:**

- The o1 models use reinforcement learning to "think" before they respond. This allows them to perform better on tasks that require reasoning as they process information.
- The o1 series introduces a new dimension in scaling, focusing on inference compute rather than just pretraining. This opens up more opportunities for growth in AI's reasoning abilities.
- While the o1 models shine in reasoning tasks, they are not always superior to GPT-4o in speed or tasks that don't require deep reasoning.

The introduction of OpenAI's o1 model series seems like a promising **development** in the world of AI. **However, some critics feel the model might be a bit overhyped.**

Here are some examples of the new model in action:

- [Coding video games from a prompt](#)
- [Math with OpenAI o1](#)
- [OpenAI o1 translates a corrupted sentence](#)
- [OpenAI o1 answers a famously tricky question for large language models](#)
- [Combining OpenAI o1 and Cursor Composer to create an iOS app in under 10 mins](#)
- [Using OpenAI o1 to create a 3D version of Snake in under a minute](#)