

L'IA en aide au développement

Max Beligné – Ingénieur de recherche
Plateforme Universitaire de Données Grenoble Alpes (PUD-GA)
max.beligne@univ-grenoble-alpes.fr

Clément Ferrandery – Stagiaire dev informatique à la PUD-GA
clement.ferrandery@gmail.com

Matériel pédagogique : <https://github.com/RondBallon/TpJ1AnfEnvol.git>

ANF Envol
Octobre 2025

Travail en binôme

Avec qui vous voulez à condition que :

- vous maîtrisiez un langage commun
- qu'un des deux ait au moins une installation fonctionnelle

(certains ont eu des difficultés à valider leur compte *Github Education*
ça serait bien qu'ils se mélangent aux autres)

Sur la partie utilisant « *VsCode + Github Copilot* »,
les deux personnes du binôme peuvent faire l'exercice en même temps
(sauf si on voit que la connexion internet a du mal)

Par contre, sur la petite partie « *Continue.dev* », important que le binôme travaille ensemble !

Objectif de la présentation

- Approfondissement de la conférence « IA4Dev » (avril 2025, Grenoble)
- Cas pratique

Important :

Cette présentation ne vise pas à promouvoir l'utilisation de l'IA pour coder
Il existe plusieurs raisons légitimes de refuser de l'utiliser
ou d'avoir un positionnement non tranché, hésitant et évoluant...

Au niveau global, dynamiques en cours importantes sur « coder avec l'IA »
Monde de l'ESR un peu en retrait

Objectifs : **informer, essayer et vous faire vos propres positionnements**

Préparation psychologique :-)

- **Même en connaissant des choses sur « code et IA »,
on peut avoir la sensation d'avoir x trains de retard**

Penser en termes de « trains de retard » n'est pas à mon avis la bonne approche.
car avec le rythme des nouveautés dans ce domaine, on aura toujours la sensation d'être en retard

Il vaut mieux se dire qu'une partie du monde de la programmation évolue,
que ça nous laisse beaucoup de choses à découvrir et à s'approprier (si on trouve ça pertinent)

- **À l'inverse, pour ceux qui veulent suivre l'actualité du domaine,
C'est bien d'avoir une veille mais attention à la fatigue sur le moyen terme**
Il y a toujours un super nouveau modèle, outil, méthode.
Impossible de tous les tester.
Il y a un « lâcher prise » à trouver ...

Une « beta présentation »

- Première fois sous ce format
- Quelques retours seraient bienvenus :
 - Points positifs
 - Points à améliorer
- N'hésitez pas à surtout développer les points à améliorer
- N'hésitez pas aussi à m'envoyer des questions

Plan

I) Etat des lieux

Histoire récente / État des usages / Choix des outils (modèles, fournisseurs, IDE et/ou agents CLI)

II) Précautions

Politiques institutionnelles et positionnement par projet
Techniques et niveaux de protection
Points d'attention liés à usage de l'IA

III) Bases (et un peu plus) pour l'utilisation

Conseils simples / Étapes pour des projets plus complexes

IV) Explorations

MCP / Développement piloté par les spécifications (Spec-Kit) / Présentation d'un agent (Open-Code)

**Avec des parties pratiques intercalées : on commencera doucement 😊
structuré sous la forme Apéro – Entrée – Plat – Dessert**

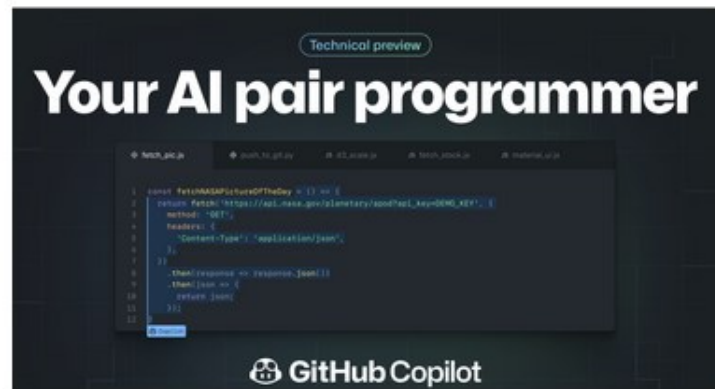
Première partie :

Etat des lieux

1) Histoire récente

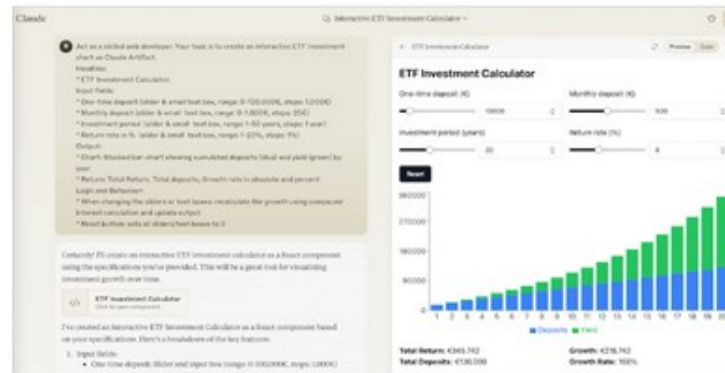
June 2021

GitHub Copilot launch introduces inline code suggestions and “pair programmer” concept



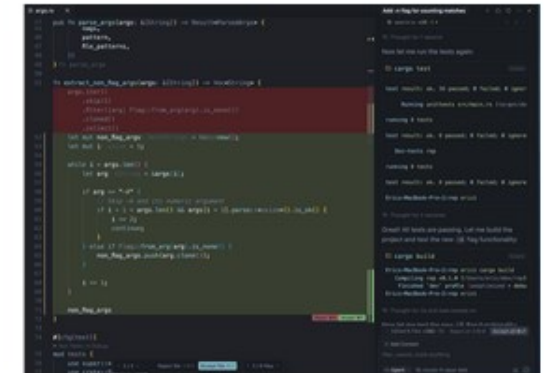
2023

From autocomplete to conversation: AI coding tools begin writing code from natural language prompts



Today

AI-native IDEs: more like a full-time engineer than assistant, writes code with minimal oversight



Extrait du rapport State of AI 2025

<https://www.stateof.ai/>

Un moment de changement: invention du « vibe coding »

Début février 2025



Andrej Karpathy ✓

@karpathy

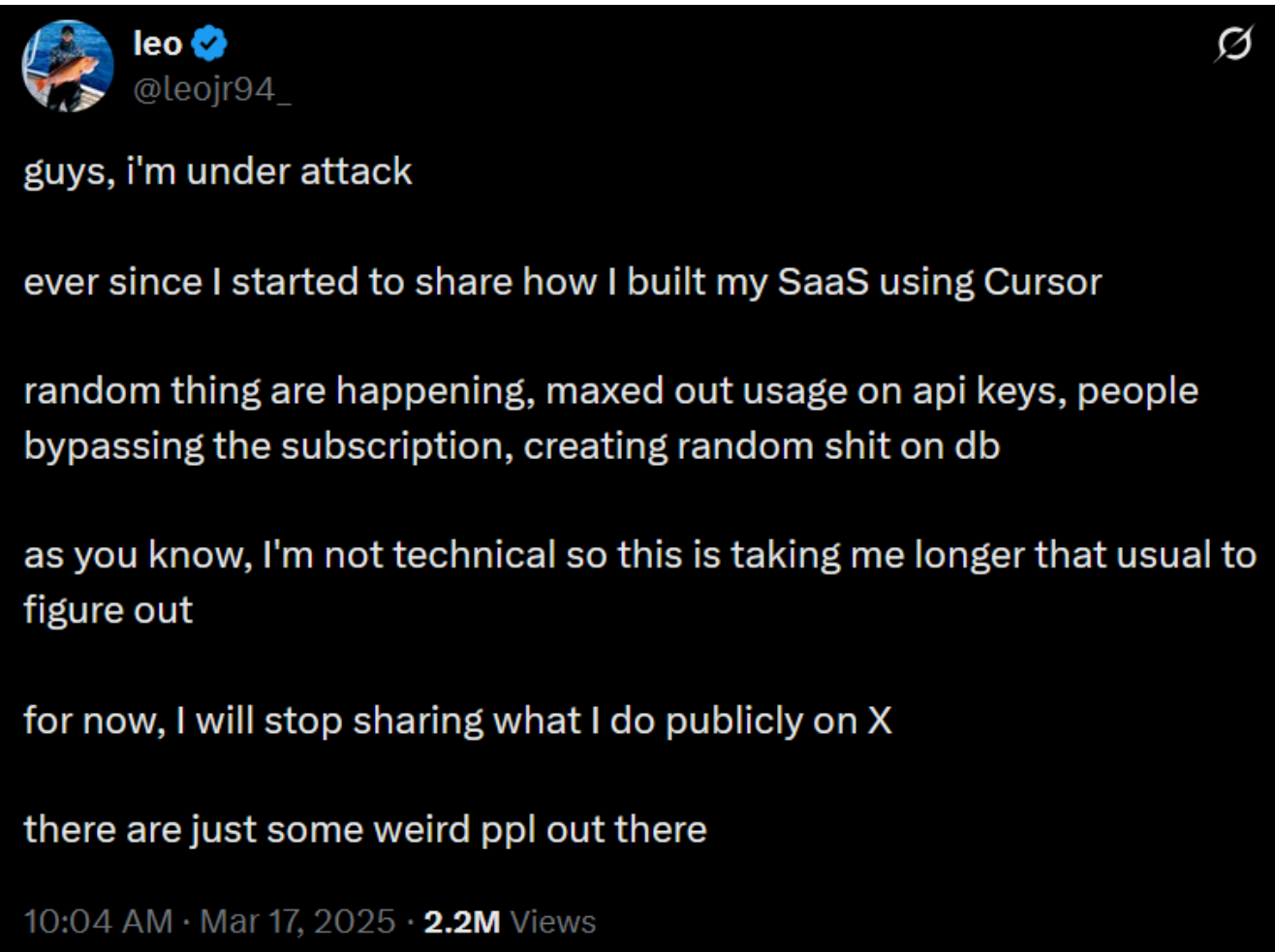


There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

Un projet « vibe codé » en production

versus

Définitions de l'ingénierie logicielle




l'ingénierie logicielle ne se résume pas à l'écriture d'un programme fonctionnel, mais à la construction d'un système capable de répondre efficacement aux besoins, de s'adapter à la demande et d'évoluer tout au long de son cycle de vie.

L'ingénierie consiste à concevoir des systèmes capables de résister aux conditions réelles, de supporter des charges de travail, de résister aux menaces de sécurité, de migrer et de prendre en charge les données des utilisateurs dans le système, et de s'adapter à de nouvelles exigences, souvent plusieurs années plus tard.

<https://serce.me/posts/2025-31-03-there-is-no-vibe-engineering>

N'est pas un monde clivé : les « vibe coders » se forment à la sécurité et les « développeurs plus traditionnels » se forment à l'IA

 u/cryptoviksant • il y a 4 j

How to ACTUALLY make your (vibe coded) apps secure (from an actual hacker)

I'm a pentester (ethical hacker) who codes SaaS part-time. I've reviewed hundreds of apps over the years, and honestly? Most have the same holes. Here's what actually keeps you safe.

- AI code review catches most issues (fr)

Look, I get it. You're shipping fast. But let Coderabbit review every pull request. It'll catch SQL injection, exposed credentials, broken auth before anything goes live. Here's a wild one: during a recent pentest, I found a race condition in a client's payment system that was double-...

Resources And Tips

620 127 2 Partager

Post sur reddit du 5 mars 2025

https://www.reddit.com/r/ChatGPTCoding/comments/1nyuh8b/how_to_actually_make_your_vibe_coded_apps_secure/

Formation ENVOL 2025 - Repenser sa façon de programmer

13–17 oct. 2025
Fuseau horaire Europe/Paris

Entrer le té

Accueil	Programme
Programme	
Inscription	Repenser sa façon de programmer

**Ne pas confondre « vibe coding »
et « codage assisté par l'IA » !**

Une approche sous forme de continuum



Codage traditionnel

Développeur :

- Ecrit toutes les lignes
- Contrôle total de tous les détails
- A priori plus lent, itération manuelle

Confiance dans l'IA

- Faible

Codage traditionnel assisté par l'IA

Développeur :

- Ecriture partielle des lignes de code, souvent assisté par l'IA
- Validation en profondeur de toutes les lignes suggérées par l'IA
- vitesse améliorée, mais nécessite toujours la supervision d'un développeur.

Confiance dans l'IA

- Moyenne

Vibe coding

Développeur :

- IA génère la grande majorité du code
- Itération rapide avec les plus souvent acceptation du code
- Le développeur est « un orchestrateur de haut niveau »

Confiance dans l'IA

- Haute

<https://gradientflow.com/vibe-coding-and-chop-what-you-need-to-know/>

2) État des usages

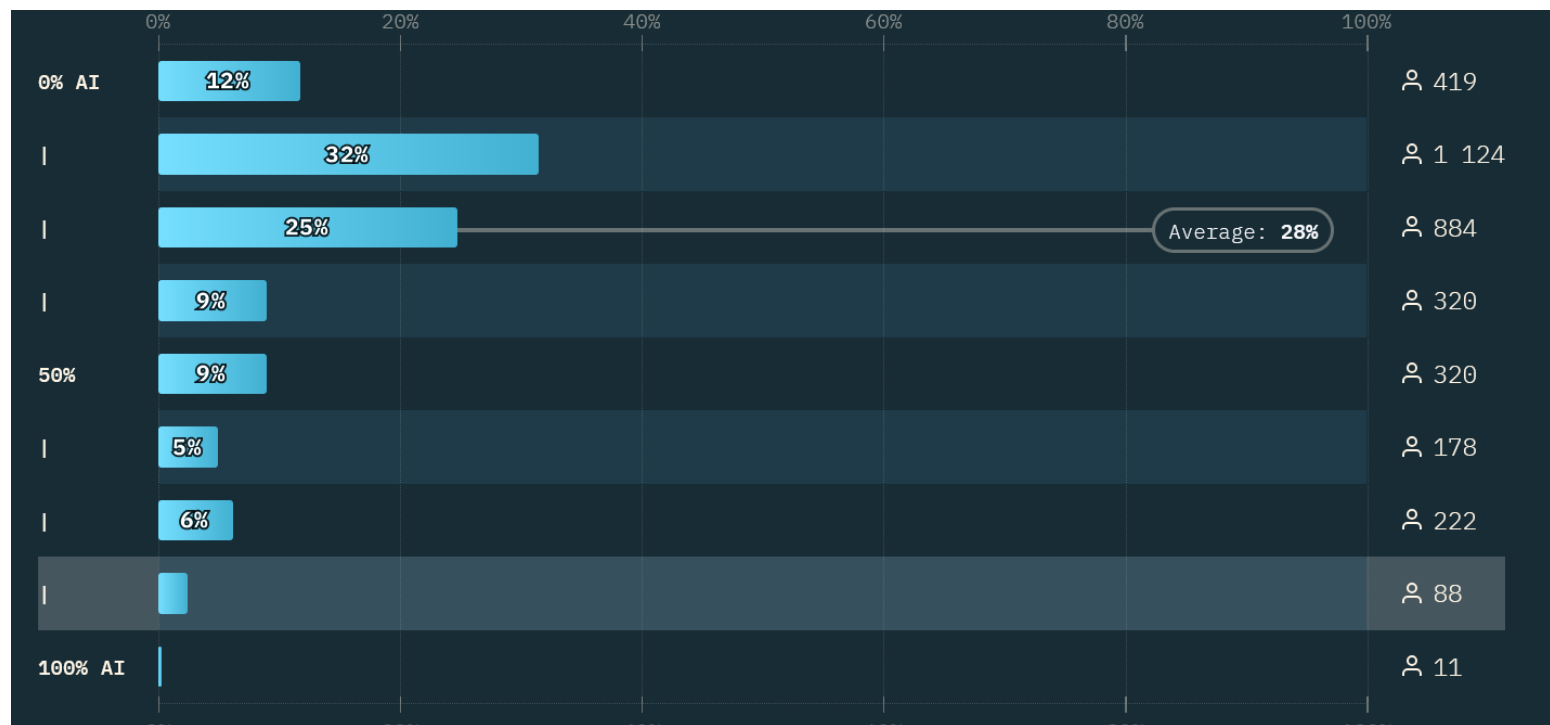
Rien trouvé spécifiquement sur l'ESR
mais beaucoup d'études sur usage général (pas toutes désintéressées :-)

- *StateOfAI 2025*, très général et très récent (9 oct) : <https://www.stateof.ai/>
Soutenu par *Air Street Capital* (société de capital-risque qui investit dans des entreprises axées sur l'intelligence artificielle)
- *State of AI-assisted Software Development* : <https://dora.dev/research/2025/dora-report/>
Google avec sa communauté *Dora*, sept 2025
- *State Of WebDev AI* : <https://2025.stateofai.dev/en-US>
Groupe de développeurs « devographics », enquête début 2025 publiée en avril

Des chiffres à regarder dans le détail

En chiffre absolu, IA beaucoup utilisée
(souvent 90 % des enquêtés mais n'est pas un échantillon tiré au hasard)

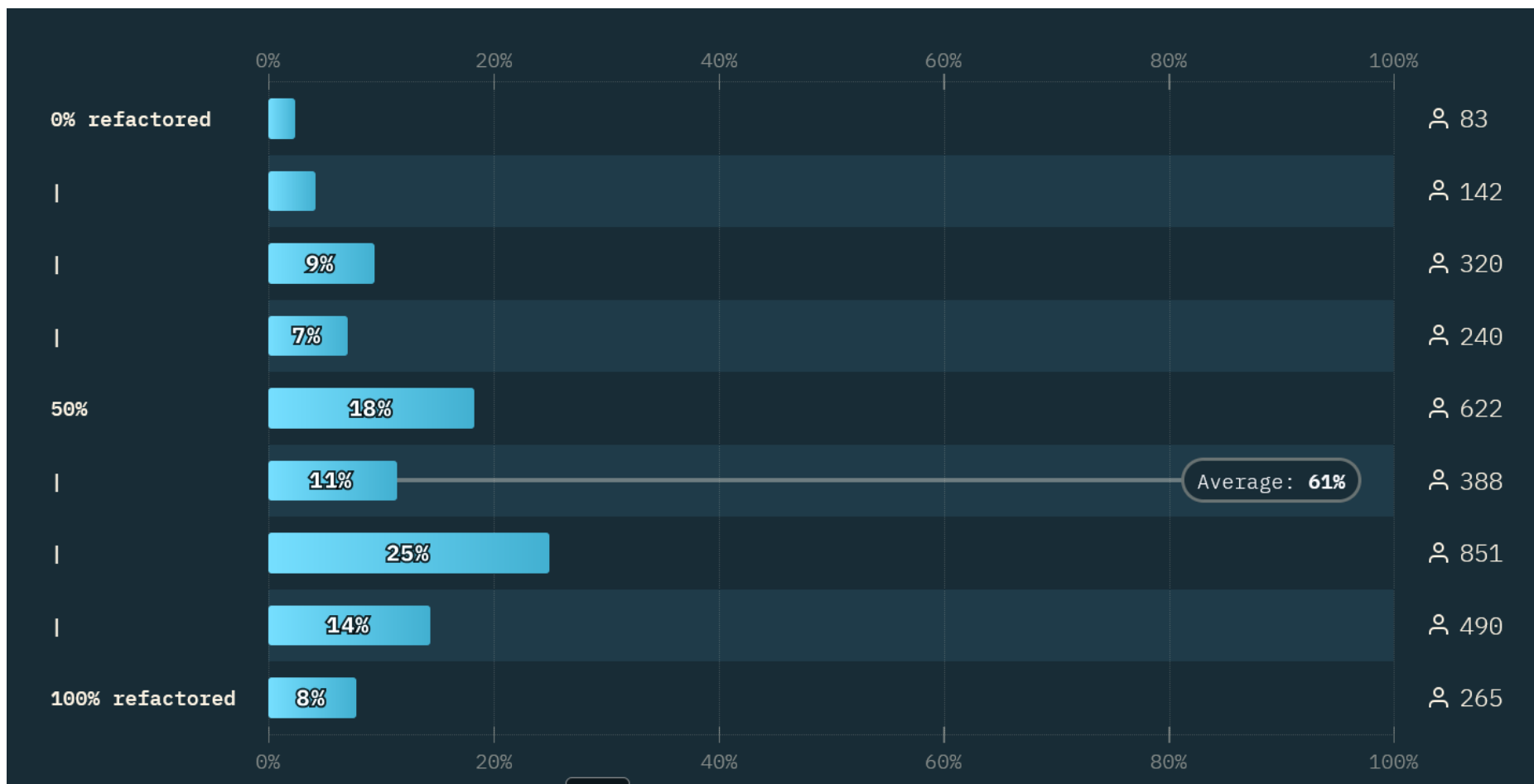
Détail *State Of WebDev AI* montre qu'on est loin d'une situation de « vibe coding »
Plus de 69 % des répondants génèrent moins de 25 % de leur code avec l'IA



Quelle proportion
de votre code est générée
par l'IA ?

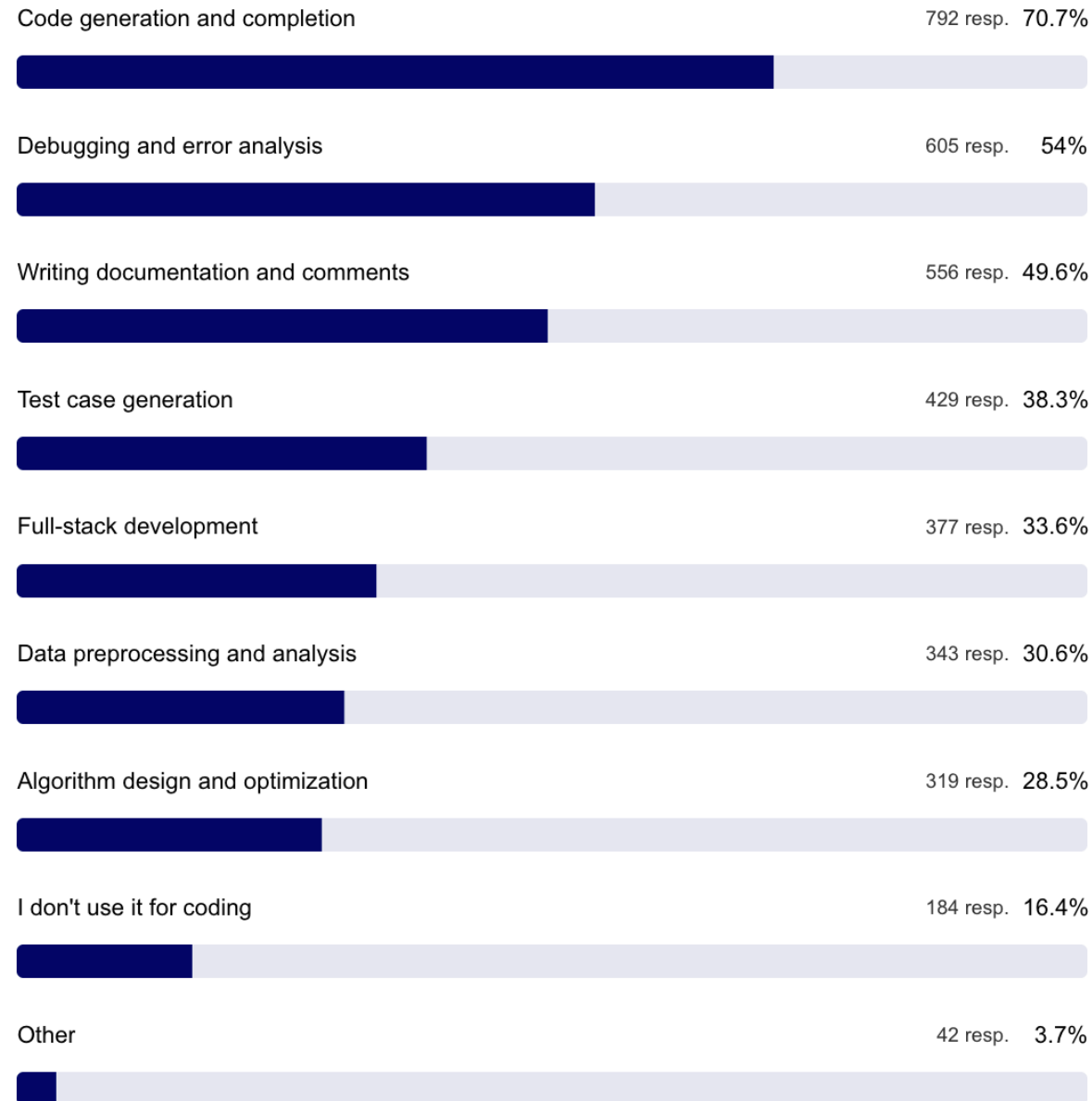
https://2025.stateofai.dev/en-US/usage/#ai_generated_code_balance

Du code largement retravaillé en aval



Lorsque vous utilisez l'IA pour générer du code, quelle proportion réécrivez-vous ou refactorisez-vous avant utilisation ?

Des utilisations multiples chez les développeurs



<https://www.stateof.ai/>

Slide 295

Diversités des usages et des fréquences

Predictive text, for example code completions	19%	17%	14%	5%	7%	22%	16%
Chat: AI-assisted conversations for coding tasks	10%	19%	19%	8%	10%	25%	9%
Collaborative: using AI to make broad, coordinated code changes	38%	20%	13%	6%	7%	12%	5%
Agent mode: AI autonomously operating in the background, possibly making changes without direct oversight	61%	12%	7%	3%	5%	8%	4%
	Never	A few times a month	1-3 days a week	4-6 days a week	Once a day	A few times a day	Hourly or more

Issu du rapport DORA 2025 : <https://dora.dev/research/2025/dora-report/>

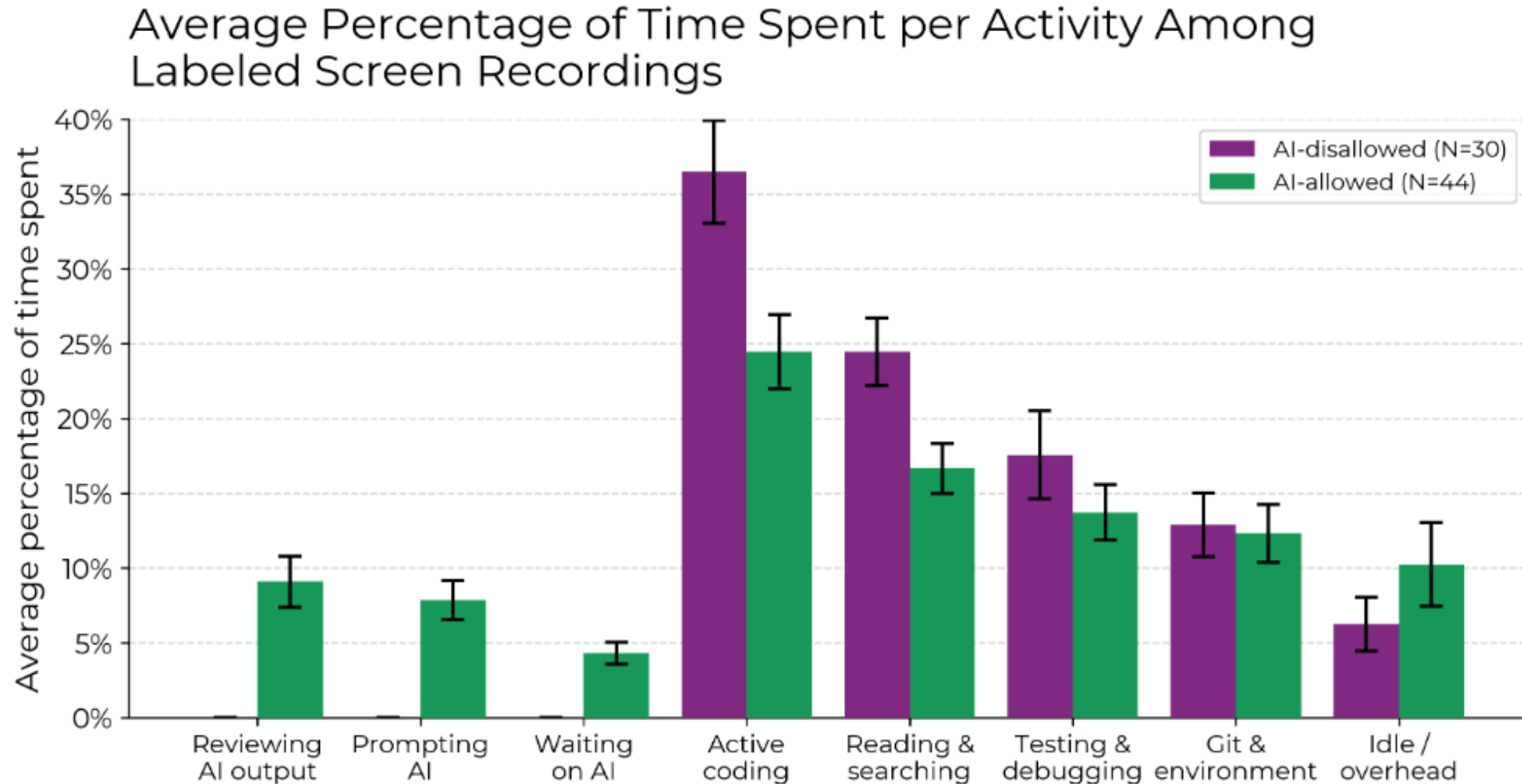
Une étude plus critique : Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (<https://arxiv.org/pdf/2507.09089>)

16 développeurs ayant une expérience modérée de l'IA réalisent 246 tâches dans le cadre de projets matures sur lesquels ils ont en moyenne cinq ans d'expérience préalable.

Chaque tâche est attribuée de manière aléatoire afin d'autoriser ou d'interdire l'utilisation des outils d'IA du début de l'année 2025 (*cursor Pro* et Sonnet 3.7/3.5)

Avant de commencer les tâches, les développeurs **prévoient que l'autorisation de l'IA réduira le temps d'exécution de 24 %**. Après avoir terminé l'étude, les développeurs **estiment que l'autorisation de l'IA a réduit le temps d'exécution de 20 %**. Étonnamment, nous constatons que **l'autorisation de l'IA augmente en fait le temps d'exécution de 19 %**

Une présentation fine de la répartition du temps dans les deux scénarios (avec IA et sans IA)



De nombreux points peuvent être discutés dans cette étude

- Étaient peu formés sur l'IDE *Cursor*
- Étaient payés à l'heure
- Évolution rapide : au moment où sort l'étude,
innovation dans l'outil qui change un peu la donne
Ex : parallélisation des tâches (juin 2025 dans *Cursor*)
permet une réduction du temps
mais demande une certaine agilité.

**Reste une étude très intéressante
car permet de remettre en question le dogme de « ça fait gagner du temps »
et pourra être reproduit dans le temps**

3) Choix des outils

Trois niveaux de questionnement :

- a) Modèle(s) utilisé(s)
- b) Fournisseur(s)
- b) IDE(s) et/ou agent(s) en ligne de commande

a) Les modèles

- Question cruciale quand on veut faire une « installation maison »

Va dépendre de vos ressources en GPU (VRAM)

Ce qu'il faut retenir, si vous avez 7Go de Vram, vous allez pouvoir faire tourner des modèles 7B

Plus un modèle est petit, plus il faut qu'il soit spécialisé pour être efficace

Pas mal de techniques pour réduire des modèles : quantification, distillation...

Au niveau technique, après c'est assez simple : *ollama*

(possibilité de surcouches *lmstudio*, etc...)


Point important, c'est la fenêtre contextuelle

(par défaut, petite sur *ollama* : 2048 token, modification avec parametre *num_ctx*
attention, augmente le besoin en VRAM avec élargissement de la fenêtre)

Le lieu pour les modèles « open-source » et « open-weight »

← → ↺ 🏠 huggingface.co/models 120 % 🌐 📧 9 📁 📄 🧑

Fichiers - Dropbox 📁 Mail 📁 Chat 📁 UGA 📁 RSocial 🧑 Helper ISSP PCS Hom...

 **Hugging Face** 🔍 Search models, datasets, users... 📦 Models 📁 Datasets 📦 Spaces 📖 Docs 💰 Pricing ⌵ | 🌐

Main **Tasks** Libraries Languages Licenses Other

🔍 Filter Tasks by name

Multimodal

🔊 Audio-Text-to-Text 🖼️ Image-Text-to-Text


📺 Visual Question Answering


📄 Document Question Answering


📺 Video-Text-to-Text 📄 Visual Document Retrieval

🧑 Any to Any


Models 2,143,945 📦 Filter by name 🔍 Full-text search ⬆️ Sort: Trending


 **neuphonic/neutts-air**
🔊 Text-to-Speech • 📄 0.7B • Updated about 14 hours ago • ⬇️ 9.24k • ❤️ 395


 **zai-org/GLM-4.6**
📄 Text Generation • 📄 357B • Updated 10 days ago • ⬇️ 21.8k • ⚡ • ❤️ 651



 **Qwen/Qwen3-VL-30B-A3B-Instruct**
🖼️ Image-Text-to-Text • 📄 31B • Updated 1 day ago • ⬇️ 98.8k • ❤️ 204


Ex d'un modèle 7B spécialisé en Rust

 **Hugging Face**

[Models](#) [Datasets](#) [Spaces](#) [Community](#) [Docs](#) [Pricing](#) 




 **Tesslate/Tessa-Rust-T1-7B-Q8_0-GGUF** 

 like 41 [Follow Tesslate](#) 371

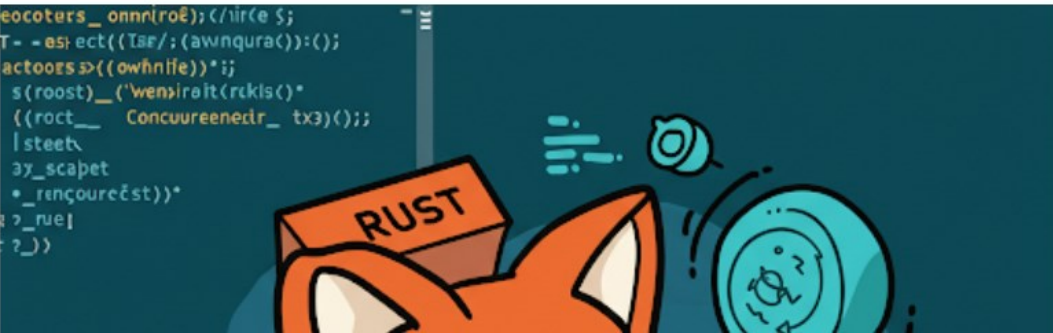
[Transformers](#) [GGUF](#) [Tesslate/Rust_Dataset](#) [llama-cpp](#) [gguf-my-repo](#) [conversational](#)

[Model card](#) [Files and versions](#) [xet](#) [Community](#) 1

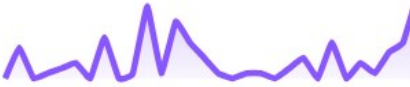
 [Train](#) [Deploy](#) [Use this model](#)



[Edit model card](#)

Tessa-Rust-T1, A Rust Focused Code Generation Model





Downloads last month
100



 **GGUF** 

[Model size](#) 7.62B params [Architecture](#) qwen2 [Chat template](#)

 Hardware compatibility [Add hardware for estimation](#) 

8-bit Q8_0 | 8.1 GB

Quels bons modèles de code généralistes « open-weight » en ce moment ?

Codestral et Devstral (24B) : <https://ollama.com/library/codestral>
<https://ollama.com/library/devstral>

Qwen 3, 30B Coder : <https://ollama.com/library/qwen3-coder:30b>

En plus gros, GML 4.6 (714 GB) et Kimi K2 (1 TB)

Pour ça, il existe des fournisseurs, exemple [Zai](#) pour GML 4,6

Va avoir un impact sur le choix de l'éditeur

Logique : « model is the product »

Question du modèle va aussi se poser quand on va utiliser *Vscode* + *Github Copilot* (300 requêtes premium avec plan éducation)

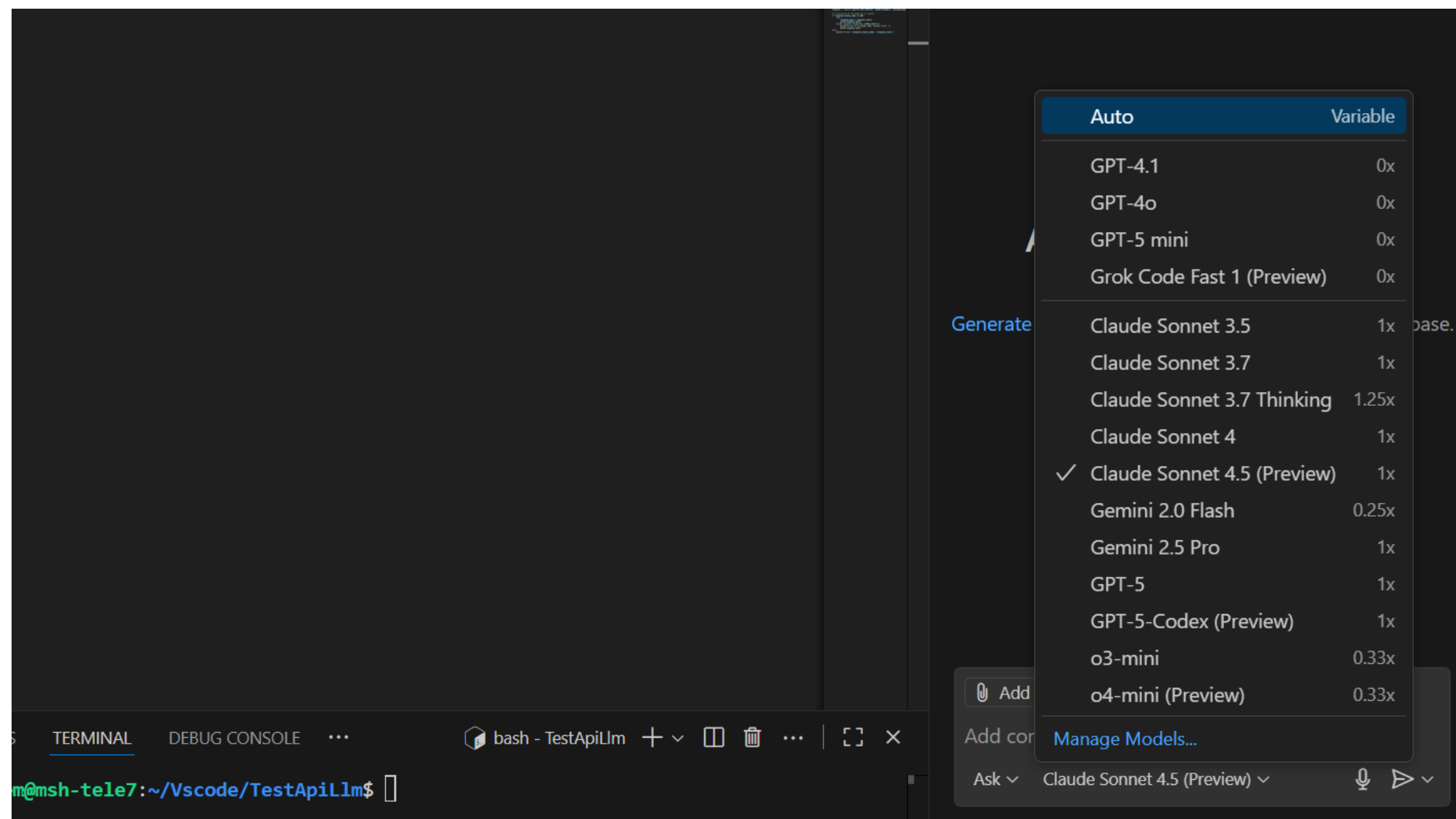
Modèle	Multiplier : « plan education »	Multiplier : « plan free »
GPT 4.1	0	1
GPT-5 Codex	1	Not Applicable
GPT-5 Mini	0	1
Claude Sonnet 4.5	1	Not Applicable
Claude Sonnet 3.7	1	Not Applicable
Claude Sonnet 3.7 Thinking	1,25	Not Applicable
Claude Sonnet 3.5	1	1
Claude Opus 4.1	10	Not Applicable
Gemini 2,5 pro	1	Not Applicable
Grok Code Fast 1	0,25	Not Applicable

Attention, il y a d'autres modèles : <https://docs.github.com/en/copilot/concepts/billing/copilot-requests#premium-requests>

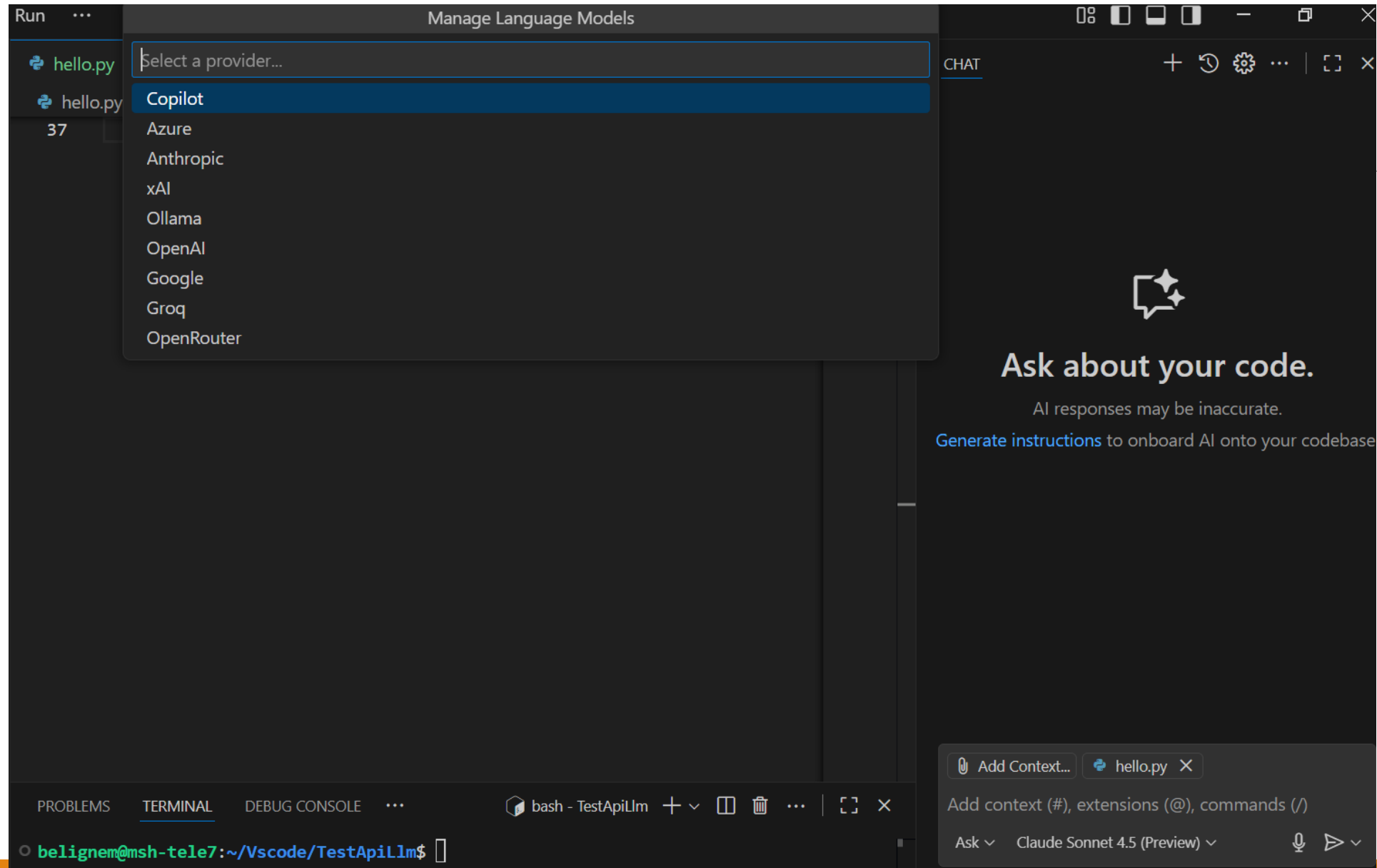
TP : « l'Apéro »

- 1) Git clone du projet si ce n'est pas déjà fait : <https://github.com/RondBallon/TpJ1AnfEnvol.git>
- 2) Ouvrir avec VsCode (code .)
- 3) Question de la mise à jour de Vscode
(désactiver temporairement car sortie juste avant le week-end de v1.105)
File > Preference > Settings
« update mode » dans la barre de recherche, puis mettre sur None
- 4) S'assurer que l'extension Chat apparaît bien pour tout le monde dans la barre latérale
(clic droit sur la barre... Si n'est toujours pas visible View > OpenView)
- 5) Passer l'extension à droite
- 6) « Disable » l'extension Continue.dev
- 7) Explorer « Modèles »

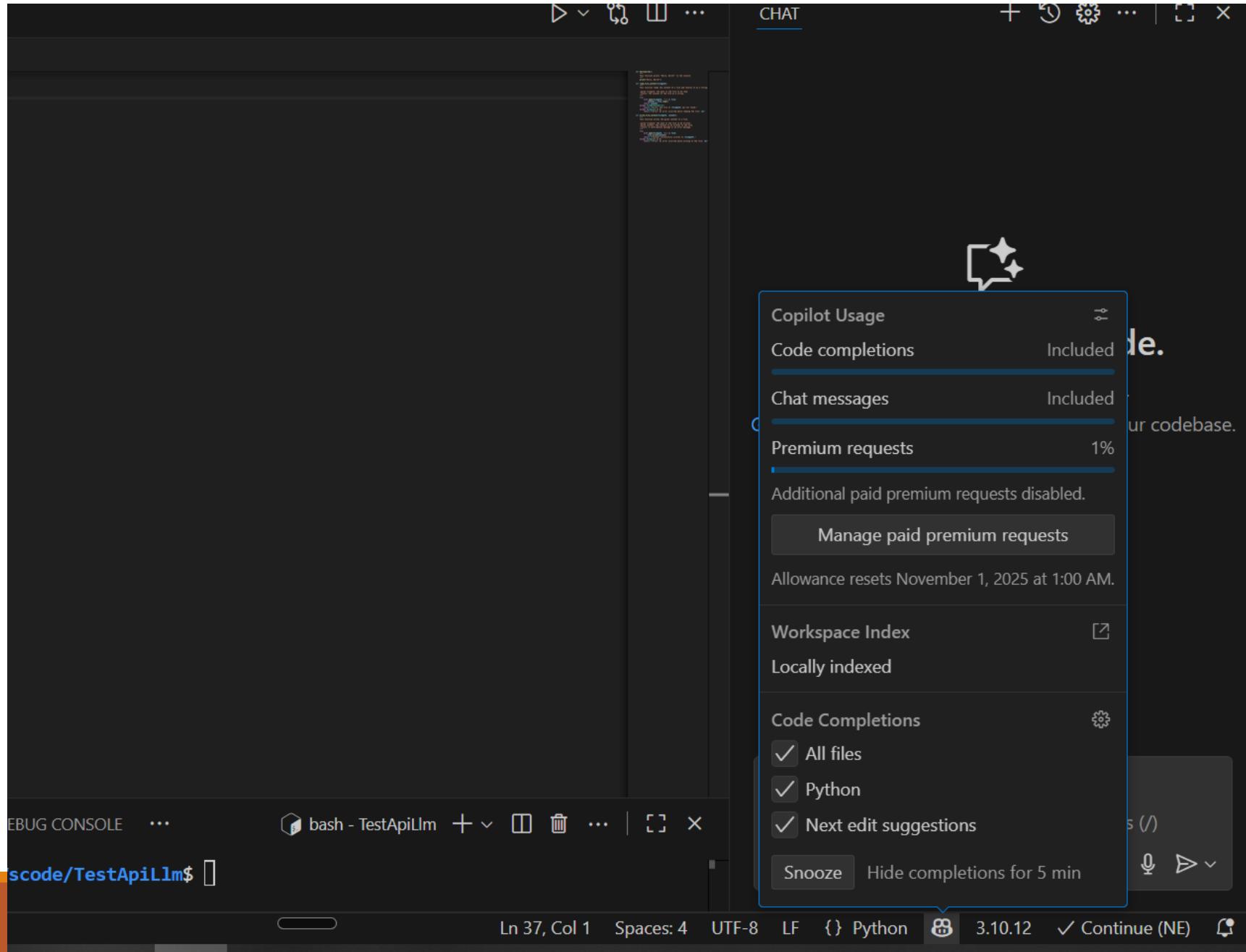
Changer de modèle dans vscode + github copilot



Possibilité « Manage modèle » pour ajout provider extérieur



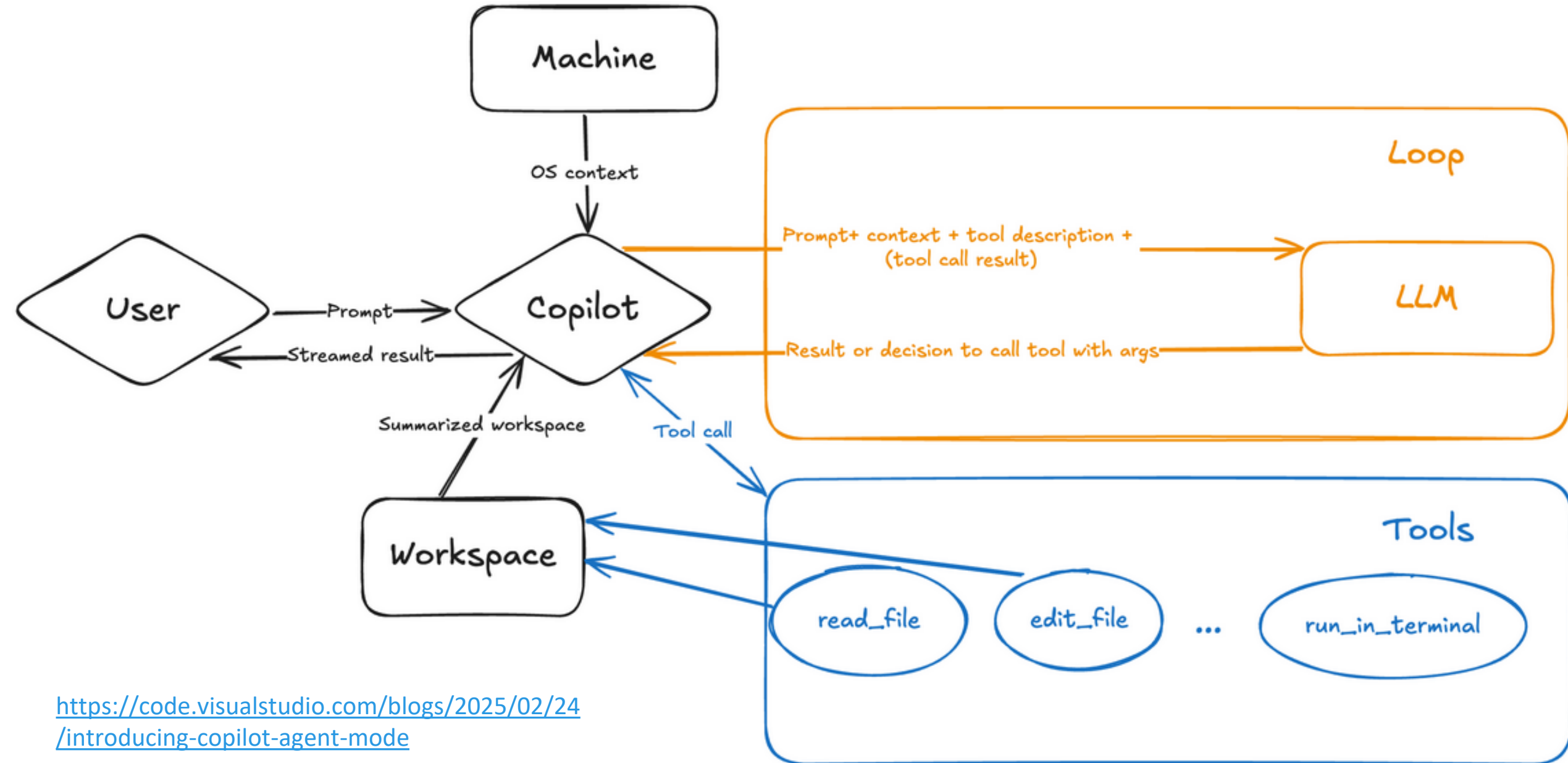
Suivre l'évolution des ses requêtes premium



Première utilisation ensemble

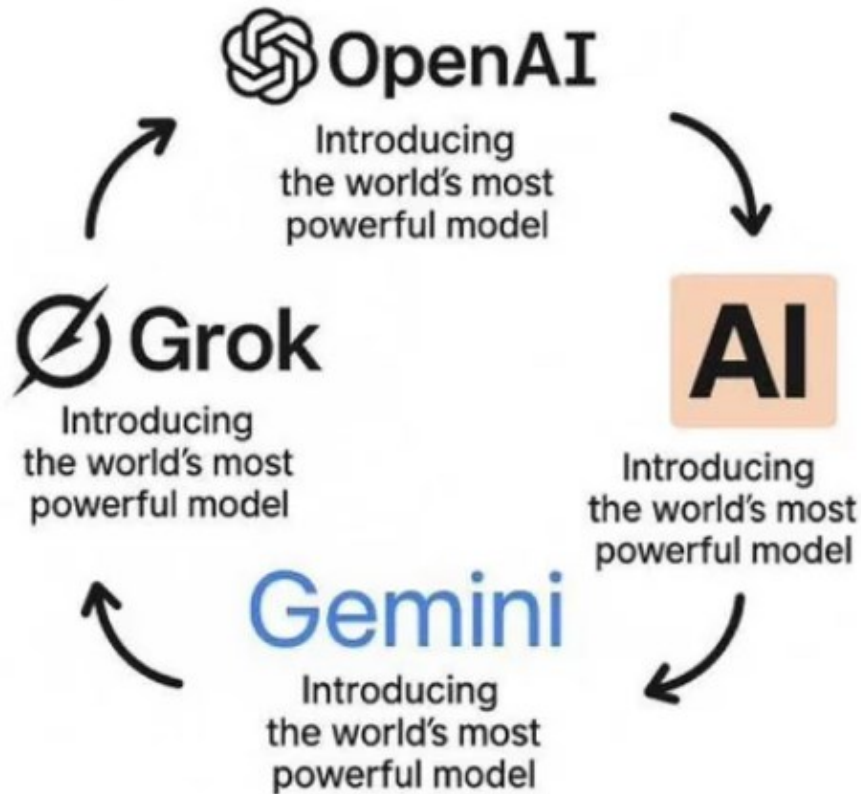
- Utilisation des modes Chat / Edit / Agent
- Ctrl + I, commande /
- Dans les prompts, pour revenir à la ligne sans valider : Alt + entrée

Slide pour expliquer le mode agent



Rappel de là où on en était avant cet apéro...

1) Etat des lieux 3) Le choix des outils a) Les modèles



En terme de performance,

- Mieux d'utiliser Claude 4.5 avec Code Claude ou GPT-5 Codex avec Codex
- Choix de *VsCode* + *Github copilot* moins optimal mais pratique

Pose aussi des questions légitimes
en matière de souveraineté + envoi données et code
(cf. partie suivante)

<https://www.reddit.com/r/OpenAI/comments/1kt8p5w/hereweagain/>

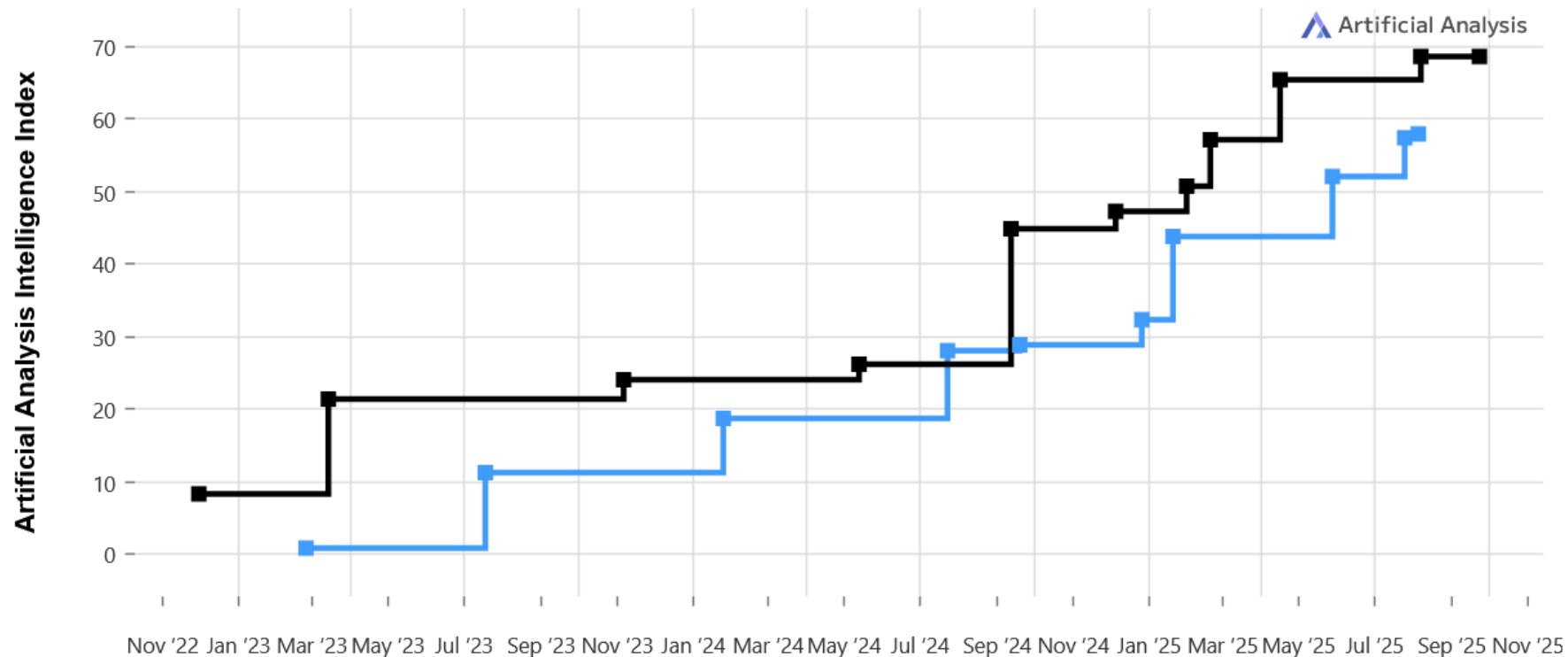
Efficacité modèle open/close source

Progress in Open Weights vs. Proprietary Intelligence



Artificial Analysis Intelligence Index v3.0 incorporates 10 evaluations: MMLU-Pro, GPQA Diamond, Humanity's Last Exam, LiveCodeBench, SciCode, AIME 2025, IFBench, AA-LCR, Terminal-Bench Hard, τ^2 -Bench Telecom

■ Open Weights ■ Proprietary



<https://artificialanalysis.ai/models/open-source>

Petit résumé des éléments importants quand on choisit un ou des modèle(s) :

- Type de données d'entraînement (des langages moins bien pourvus)
- Taille du modèle
- Taille de la fenêtre contextuelle
(**Attention ici, en lien aussi avec l'IDE 128k dans VS Code Classique**
200k de contexte dans VS Code Insider
Est obligé sur pas mal de tâches de faire des « résumés de contexte » : perd en efficacité)
- Spécialisation
- Localisation de l'exécution du modèle (serveurs distants ou en local)
- Modèles propriétaires ou «open source »

b) Les fournisseur(s)

Va dépendre du modèle choisi.

Quand vous utilisez *Vscode* + *GitHub copilot*, c'est une étape qu'on a tendance à oublier mais si jamais vous voulez sortir de ce système, c'est une question qui va se poser

Résumé rapide de ce qui existe autre que les plus connues *Antropic*, *OpenAi*, *DeepSeek*, *Zai*

- En commercial avec des modèles gratuits : <https://openrouter.ai/>
- Se rapprocher de vos établissements :
Est-ce qu'ils ont une instance *OpenWebUI* que vous pouvez utiliser
- **API Alber** : <https://albert.sites.beta.gouv.fr/>
cf. partie sur *Continue.dev* pour l'intégration



ILaaS : une fédération d'universités pour mutualiser les ressources GPU

<https://www.ilaas.fr/>

En fonction des ressources et des requêtes, dispatcher en amont
Permet de proposer un spectre de modèles plus important de manière plus stable (redondance)

Etablissements de départ : Rennes, Reims, Lille, Paris 1 Panthéon-Sorbonne, Université de Lorraine, Centrale-Supelec

Etablissement en cours ou en discussion : Strasbourg, La Rochelle, Nantes, Toulon, Angers, Côte d'Azur, UPJV – Amiens, UTC, Cergy, AMU, Orléans

Une belle dynamique !
contact: O Wong à Rennes

c) IDE(s) et/ou agent(s) en ligne de commande

Par rapport à la démonstration *Vscode* (IDE),
il existe aussi des agents tout en ligne de commande (*Claude Code*, *Codex...*)

Possibilité d'allier l'IDE et l'agent en ligne de commande
(Vont surement d'ailleurs de plus en plus se connecter entre eux via le protocole ACP
cf. Zed avec Claude Code)

Sur le sujet des IDE, vraiment beaucoup de choix
Peu d'expérience (j'ai surtout utilisé *Vscode*) et peu d'avis
à part que ça serait intéressant d'avoir des licences pour *MistralCode* dans l'*ESR*

Deuxième partie :

Précautions

1) Politiques institutionnelles et positionnement par projet

Préalable et veille à avoir : chercher le ou les positionnement(s) de votre ou vos établissement(s) et tutelle(s) sur le sujet ?

Pas forcément facile de trouver et très explicite ...

Exemple publication récente : collaboration *DevLOG*, la *DAJ CNRS* et *CNRS innovation*
Plaquette [JE CODE : QUELS SONT MES DROITS ? QUELLES SONT MES OBLIGATIONS ?](#)

- Très peu de choses sur l'IA
- renvoie à l'intervention du service juridique de l'UGA à la journée IA4Dev
qui dit en gros « c'est le farwest », « appel au bon sens », « on est en retard sur ce sujet »

Envoi de mails pour obtenir le positionnement du *CNRS* sur le sujet

- Pour l'instant, est resté sans réponse

Que faire si vous n'avez aucune info, le cadre qui s'applique, c'est le RGPD et l'IA Act

Exemple de l'Inrae

Document de référence : <https://science-ouverte.inrae.fr/fr/offre-service/fiches-pratiques-et-recommandations/utiliser-les-ia-generatives-comme-assistant-personnel-au-sein-dinrae>

Approche du document en terme d'opportunités, de risques et de recommandations

Reste à un niveau très général

→ *in fine*, vous savez qu'il faut faire attention
mais reste une grande part de flou

Peut et doit être discuté au niveau du projet :

Exemple perso : Début d'un travail sur une codebase en partie déjà créée

Demande à l'auteur du code :

« Est-ce qu'on peut utiliser *github copilot* (microsoft)
en envoyant suivant le modèle utilisé des parties de ton code à *Antropic* ou *OpenAi* ? »

Réponse :

« Honnêtement je me sers un peu de copilot pour coder et il n'y a rien de super innovant dans mon code. Pas de souci donc... du moins tant que ce ne sont que des bouts de code et pas l'architecture d'ensemble »

Là, discussion en cours de projet

Mais d'après moi, va de plus en plus être défini en début de projet :

quel(s) positionnement(s) par rapport aux IA propriétaires et celles *open source* installées localement

➡ **Va dépendre des projets, de leurs enjeux, des acteurs et des institutions qui les portent**

Exemple de recommandation plus précises et strictes

Recommandations de sécurité pour un système d'IA générative (ANSSI, 2024)

R30

Contrôler systématiquement le code source généré par IA

Le code source généré par IA doit faire l'objet de mesures de sécurité afin de vérifier son innocuité :

- proscrire l'exécution automatique de code source généré par IA dans l'environnement de développement ;
- proscrire le *commit* automatique de code source généré par IA dans les dépôts ;
- intégrer un outil d'assainissement de code source [3, 4] généré par IA dans l'environnement de développement ;
- vérifier l'innocuité des bibliothèques référencées dans le résultat du code source généré par IA ;
- faire contrôler régulièrement par un humain la qualité du code source généré à partir de requêtes types suffisamment sophistiquées.

Exemples outils d' « assainissement » du code

2 types :

- **Multi-langages** (Python, Java, Js, C/C++) : *SonarQube, Semgrep ...*
- **Outils spécialisés dans un langage** : ex *Bandit* pour python

Certains peuvent s'installer comme extension dans *Vscode*

Pour ceux qui ne connaissent pas *Vscode*,
monde important des extensions

R31

Limiter la génération de code source par IA pour des modules critiques d'applications

Il est fortement recommandé de ne pas utiliser un outil d'IA générative pour générer des blocs²² de code source destinés à des modules critiques d'applications :

- les modules de cryptographie (authentification, chiffrement, signature, etc.);
- les modules de gestion des droits d'accès des utilisateurs et administrateurs;
- les modules de traitement de données sensibles.

R32

Sensibiliser les développeurs sur les risques liés au code source généré par IA

Il est recommandé d'effectuer des campagnes de sensibilisation sur les risques liés à l'utilisation de code source généré par IA. Cette sensibilisation peut s'appuyer sur des rapports publics sur ce sujet ou bien des papiers de recherche²³ démontrant la présence de vulnérabilités dans le code généré par IA.

En complément, les développeurs peuvent également être formés sur les outils d'IA pour l'optimisation de leurs requêtes (*prompt engineering*²⁴) afin d'améliorer la qualité et la sécurité du code généré.

2) Techniques et niveaux de protection

a) Protection des secrets

Enquête [The state of secrets sprawl 2025](#) (gitguardian)

Fait le constat d'une augmentation des problèmes de divulgation des secrets (clé API, jeton...) avec les codes générés par l'IA (4,6 → 6,4 % des projets)

2 causes d'après l'étude :

- Le code généré par l'IA peut être intrinsèquement moins sécurisé
- L'utilisation d'assistants de codage peut pousser les développeurs à privilégier la productivité au détriment de la qualité et de la sécurité du code

Beaucoup d'autres problèmes:

- Pb des clés dans les repos privés avec attaque (concerne github) : <https://korben.info/camoleak-github-copilot-vulnerability-prompt-injection.html>
- Beaucoup de secrets pas mis sur Github (grâce a .gitignore) mais envoyés par API à ou aux assistants IA

Illustration du problème

- 1) Créer un dossier : « TestCopilot » et l'ouvrir dans Vscode
- 2) Créer un fichier .env
- 3) Taper APIKEY=12

Objectif : éviter qu'il y ait de l'auto-complétion dans un .env
et toutes ses variations .env.local, .env.development, .env.production

1^{ère} solution au niveau de Vscode

Ctrl+Shift+P puis **“Preferences: Open User Settings (JSON)”**

Descendre jusqu'à : `"github.copilot.enable": {`
 `"*": true,`
 `"plaintext": false,`
 `"markdown": false,`
 `"scminput": false },`

Traduction : `"*": true` signifie que par défaut tous les fichiers sont activés pour github copilot
 `"plaintext": false` signifie que les fichiers plaintext sont exclus de copilot

On va ajouter un `".env*": false`, après la ligne `"*": true`,
(Si vous avez plusieurs .env comme .env.development .env.production....
vont tous être compris dans l'exclusion)

Exemple plus évolué

de Paul Sorensen (<https://paulsorensen.io/github-copilot-vscode-privacy/>)

```
"files.associations": {  
  ".env*": "dotenv",  
  "*.cfg": "ini",  
  "*.cnf": "ini",  
  "*.conf": "ini",  
  "*.config": "ini"  
},  
"github.copilot.enable": {  
  "*": true,  
  "dotenv": false,  
  "ini": false,  
  "json": false,  
  "yaml": false  
},
```

2^{ème} solution au niveau d'un projet

Vous n'avez pas envie de créer une exception au niveau global

Créez un fichier `.vscode/settings.json` à la racine de votre projet

Puis spécifiez dedans

```
{  
  "github.copilot.enable": {  
    "*": true,  
    "Data/ANePasPartager.csv": false  
  }  
}
```

b) Protection de vos données et codes

- Etales** :
- 1) Se connecter à votre compte *Github*
 - 2) Cliquer sur votre image de profil
 > *Copilot Setting*

Vérifier : - que la recherche internet est désactivée
 - que Github n'utilise pas vos données pour améliorer ses produits
 - que Github n'utilise pas vos données pour entrainer ses modèles d'IA

Se protéger aussi à partir de Vscode

Plusieurs paramètres de télémétrie liés à *Vscode*

Détaillés par Paul Sorensen : <https://paulsorensen.io/github-copilot-vscode-privacy/>

Intégrer sa proposition de settings.json à vos settings suivant vos choix
(Je les ai testés et mis sur le git dans le dossier « Ressource »)

Attention : Si vous avez du code sensible ou des données RGPD,
même en réalisant toutes les étapes précédentes, n'utilisez pas *Github Copilot*
Adaptez vous en fonction de vos projets, de vos positions perso, projets et institutions

c) Problème de la propriété intellectuelle

Problème si *Github Copilot* vous suggère du code proche d'un code public avec licence

Setting : Suggestions matching public code

Envoi des petits extraits de code pour comparaison avec grosse base de données recensant code non libre de droit.

Là, il y a un choix à faire :

- soit vous voulez rien n'envoyer
mais vous ne saurez pas si vous avez du « code public » sous licence dans vos suggestions
- soit vous envoyez et *Github Copilot* vous avertit à chaque fois que c'est le cas

3) Points d'attention liés à usage de l'IA

APPS BUILD WITH
NO AI IN 5 HOURS



APPS BUILD WITH
AI AGENTS IN 5 MIN



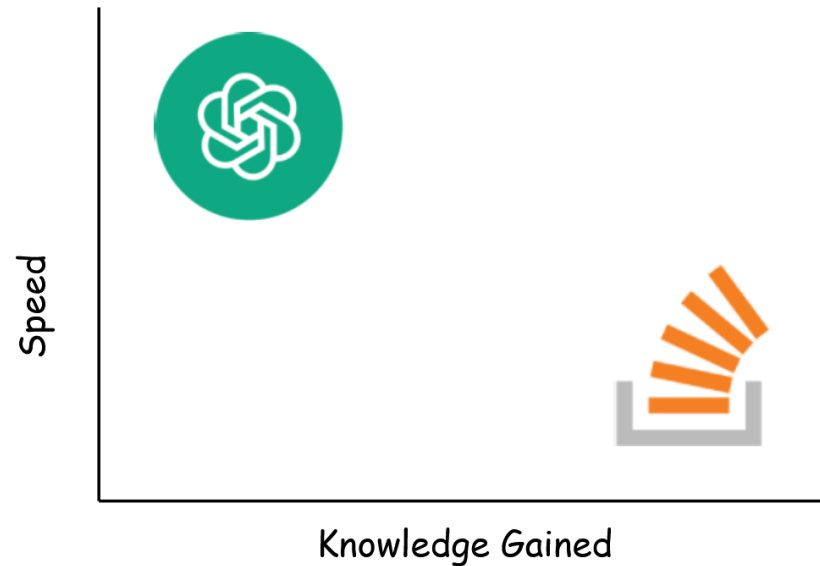
Besoin de : - simplifier

- découper en petites étapes**
- de vérifier au fur et à mesure**
- de tester au fur et à mesure**

<https://www.codeflash.ai/post/llms-struggle-to-write-performant-code#the-complex-art-of-performance-optimization>

Le piège de la vitesse

Schéma un peu caricatural mais met en avant une problématique réelle

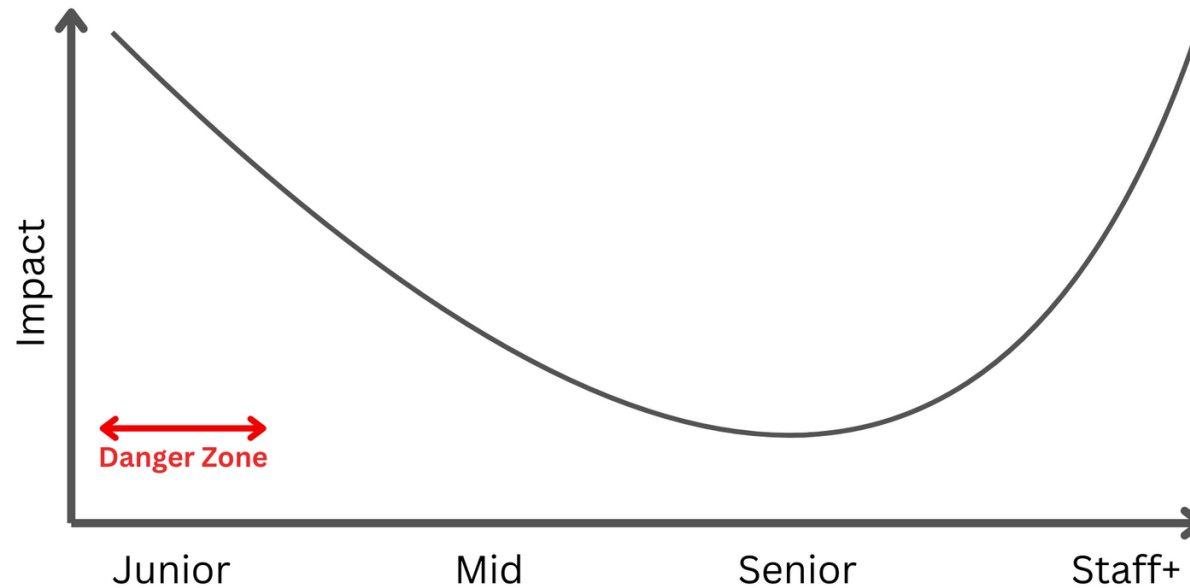


<https://nmn.gl/blog/ai-and-learning>



Est-ce qu'on profite de l'IA pour en faire toujours plus et toujours moins bien ?
ou est-ce qu'on ré-alloue le temps gagné à des choses pertinente, comme la compréhension en profondeur,
la consolidation des connaissances et l'acquisition de nouvelles compétences ?

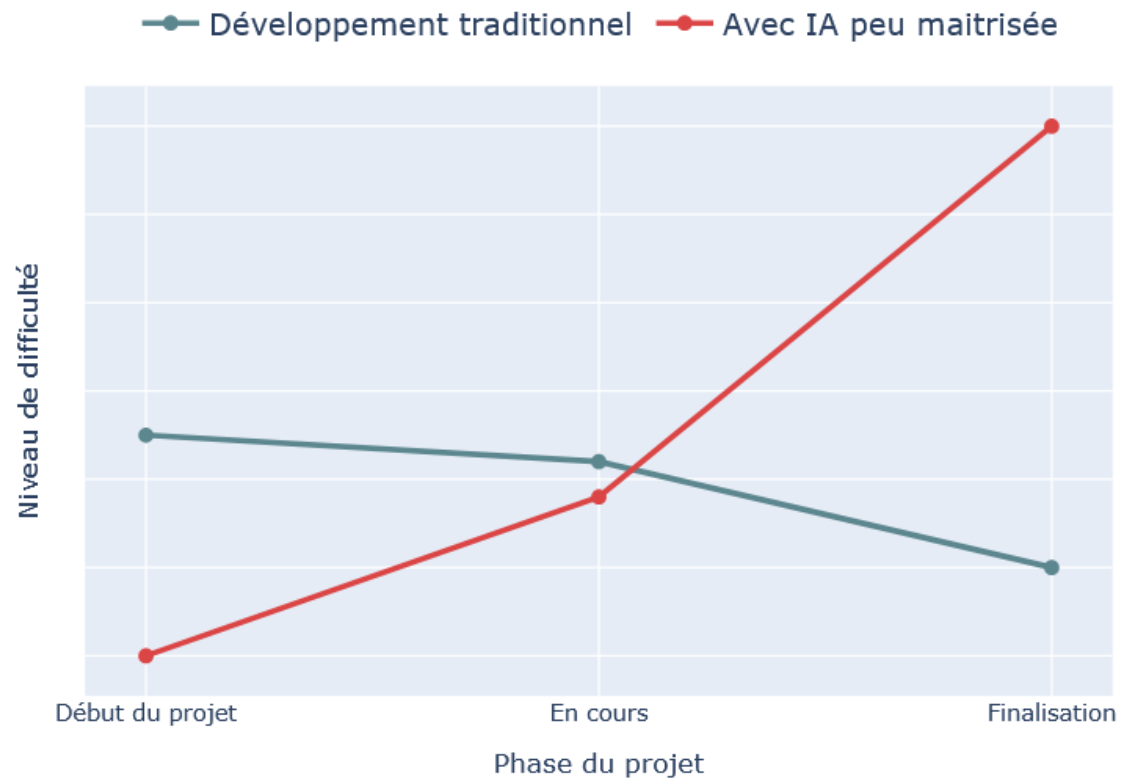
Débutants : une attention particulière !



Source : <https://serce.me/posts/2025-02-07-the-ilm-curve-of-impact-on-software-engineers>

L'empilement des POC

Difficultés des projets de développement



Avec possiblement une pression externe ([site](#))



« Souvent sous la pression de leur direction générale, les DSI se retrouvent noyés dans une multitude de PoC à base de GenAI ».

(Photo : Alexandra Koch/Pixabay)

L'effet GPS (dépendance et possible perte de compétence)

Diminution progressive du temps avant lequel on a recours à l'IA

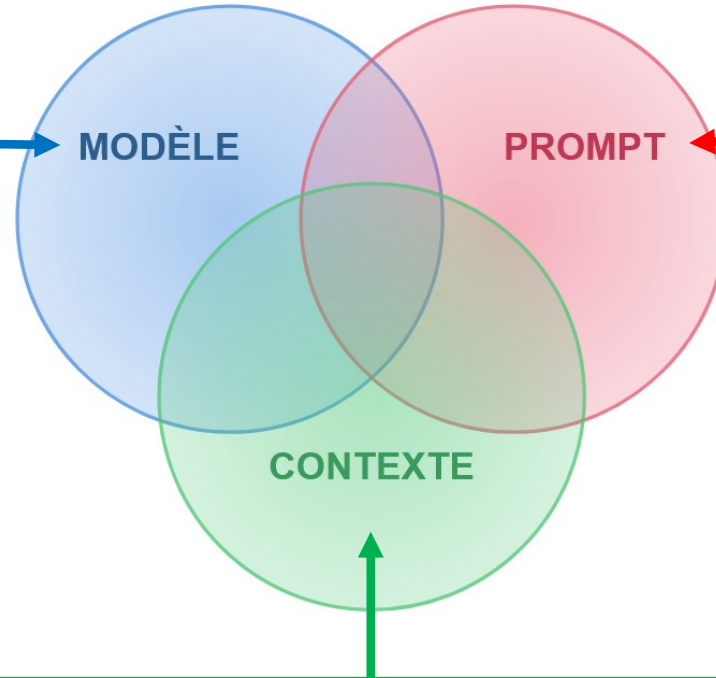
face à un problème : 1h → 30 mn → 10 mn → 1 mn → 10 s → 1 s

Troisième partie :
Bases et un peu plus
pour l'utilisation

1) Conseils au plus simple

Bien choisir mode chat / edit / agent + trilogie ci-dessous

- GPT 4.1
pour tâche simple
- Sonnet 4 ou 4.5
gpt 5 ou gpt 5-codex
pour tâche complexe



- Essayez d'être précis
- Itérez : regardez ce que l'IA n'a pas compris par rapport à ce que vous vouliez, expliquez lui mieux et ré-essayez

Associez à votre demande les fichiers nécessaires (#) pour la comprendre
Si ce sont des données avec pb RGPD, créez des fausses données similaires
Si long fichier de code essayez de refactoriser,
si long fichier de données et pas nécessaire de tout envoyer, faire un résumé de la structure ou petit ex

TP : « L'entrée » en autonomie

Un fichier CSV avec id, nom, prénom, âge, profession

Un fichier Excel avec liste des paires « professions <-> Code PCS »

→ On veut obtenir un csv avec id, nom, prénom, âge, profession, code PCS pour chaque individu du fichier initial

Essayez déjà avec *Github Copilot*

Puis refaire l'exercice avec *Continue.dev*

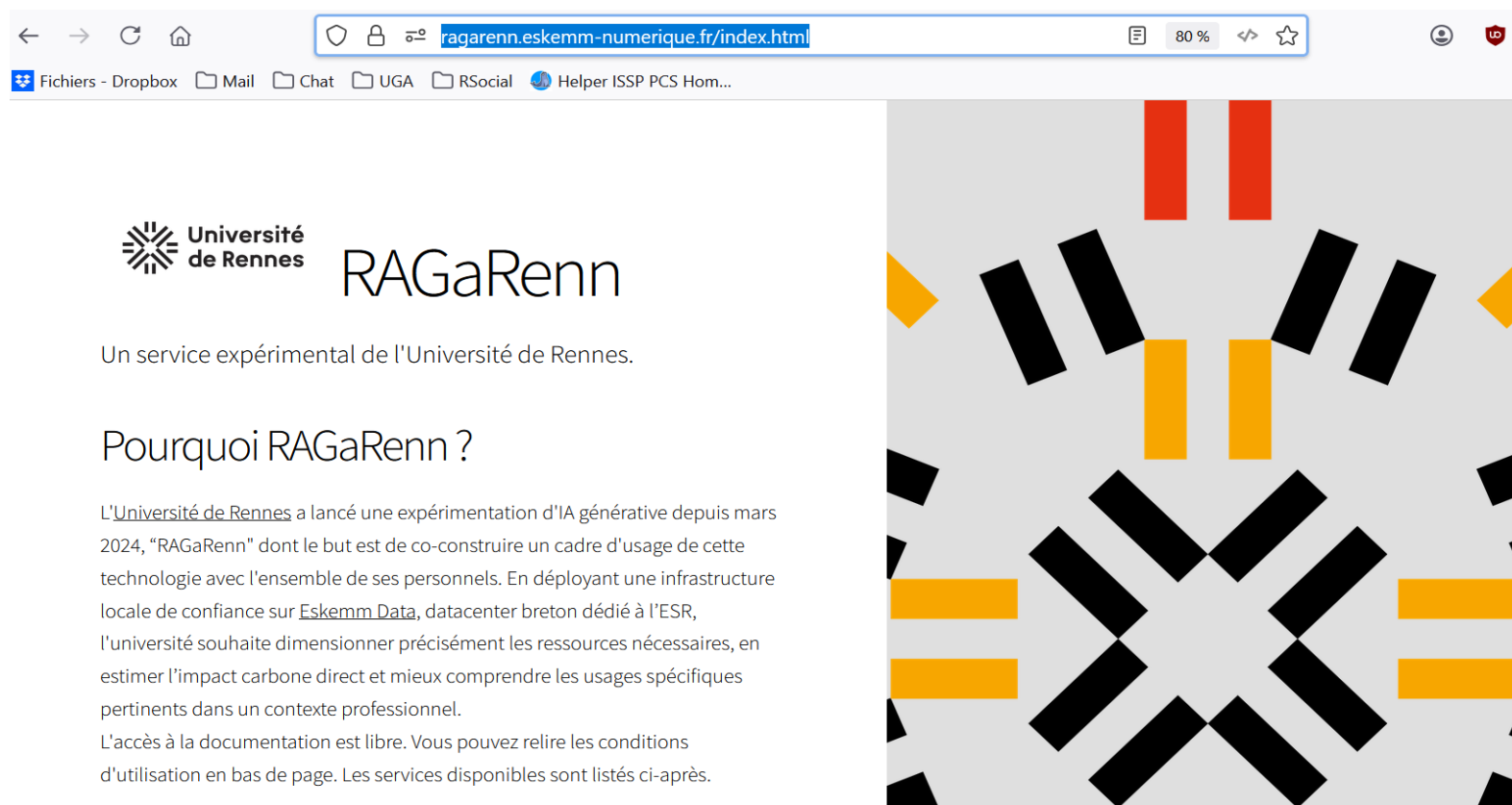
(Cf. les étapes suivantes pour *Continue.dev*)

Continue.dev nécessite d'avoir un fournisseur

On va utiliser le service OpenWebUI de [RagARenn](https://ragarenn.eskemm-numerique.fr/index.html)

A été gentiment mis à disposition pour la formation.

Pour vos besoins quotidiens, voir avec vos structures si elles ont un service similaire existant !



Accepter les conditions d'utilisation

Service disponible > ESR > Authentification avec fédération d'entités Renater



The screenshot shows a web browser window with the URL `ragarenn.eskemm-numerique.fr/sso/ch@t/app`. The browser's address bar and tabs are visible at the top. Below the browser, the Mistral AI interface is shown. On the left, a sidebar contains a hamburger menu icon and the text "Mistral Small 3.1 24B" with a dropdown arrow and a plus sign. Below this, it says "Définir comme valeur par défaut". On the right side of the interface, there are three icons: a list icon, an edit icon, and a file upload icon. The file upload icon, which shows a document with a green checkmark, is circled in red, and a red arrow points to it from the bottom right. The main content area features the Mistral logo (a stylized sunburst) followed by the text "Mistral Small 3.1 24B" and the subtitle "Modèle généraliste très rapide". Below this is a large text input field with a plus sign on the left and microphone and headset icons on the right. The input field contains the text "Comment puis-je vous aider aujourd'hui ?". Underneath the input field, there is a "Suggéré" (Suggested) section with a lightning bolt icon. It contains a suggestion box with the text "Give me ideas" and "for what to do with my kids' art".

← → ↻ 📄 ragarenn.eskemm-numerique.fr/sso/ch@t/app 📄 ☆ 📄 | 👤 ⋮

📄 | 📧 Gmail 📺 YouTube 📍 Maps

☰ **Mistral Small 3.1 24B** ▾ +
Définir comme valeur par défaut

☰ ✎ 📄

 **Mistral Small 3.1 24B**
Modèle généraliste très rapide

+ Comment puis-je vous aider aujourd'hui ? 🎤 🎧

⚡ Suggéré

Give me ideas
for what to do with my kids' art

On va aller récupérer une clé API en allant dans les paramètres

≡ Mistral Small 3.1 24B ▾ +

Définir comme valeur par défaut



⚙ Paramètres

🗑 Conversations archi...

🔗 Déconnexion

● Utilisateurs actifs: 6



Mistral Small 3.1 24B

Modèle généraliste très rapide

+ Comment puis-je vous aider aujourd'hui ?



⚡ Suggéré

Give me ideas

for what to do with my kids' art

Paramètres



🔍 Recherche

⚙️ Général

💻 Interface utilisateur

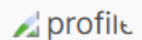
👤 Personnalisation

🔊 Audio

💬 Conversations

➡️ **👤 Compte**

ℹ️ À propos



profile

Image de profil

Utiliser les initiales

Utiliser Gravatar

Retirer

Nom d'utilisateur

Max BelignÃ©

Webhook de notification

Entrez l'URL de votre webhook

Changer le mot de passe

Afficher

➡️ **Clés d'API**

Afficher

Enregistrer

Paramètres

🔍 Recherche

⚙️ Général

💻 Interface utilisateur

☁️ Connexions

👤 Personnalisation

🔊 Audio

💬 Conversations

👤 Compte

ℹ️ À propos



Image de profil

Utiliser les initiales

Utilisez Gravatar

Retirer

Nom

beligne max

Notification Webhook

Enter your webhook URL

Changer le mot de passe

Montrer

Clés d'API

Cacher

Jeton JWT

..... 🔍 📋

Clé d'API

+ Créer une nouvelle clé secrète

Paramètres



Recherche

Général

Interface utilisateur

Personnalisation

Audio

Conversations

Compte

À propos

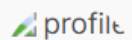


Image de profil

Utiliser les initiales

Utiliser Gravatar

Retirer

Nom d'utilisateur

Max BelignÃ©

Webhook de notification

Entrez l'URL de votre webhook

Changer le mot de passe

Afficher

Clés d'API

Cacher

Token JWT

.....



Clé d'API

.....



Enregistrer

Continue.dev



Assistant de code open-source basé sur l'IA

Présentation : <https://www.continue.dev/>

Nous, on va l'utiliser par l'extension VsCode

(Normalement, vous avez déjà dû l'installer

Sinon, il faudrait le faire mais à voir en fonction de la connexion internet)

Gestion du passage de l'utilisation de *Copilot* à *Continue.dev*

Conseils : - Disable extension *Github Copilot* (cf. slide suivante) et l'enlever du panneau de droite
(Attention si vous êtes sous WSL, dans les extensions, il y a deux zones local et WSL
Faire le disable sur WSL)

- Enable extension *Continue.Dev* et la mettre sur le panneau de droite
(Sous WSL, bizarre car n'apparaît pas sur la zone WSL mais marche quand même)

Car il peut avoir des problèmes d'interférence entre *Copilot* et *Continue.dev*
(notamment sur le Ctr + I)

EXTENSIONS

Search Extensions in Marketplace

LOCAL - INSTALLED 25

Install in WSL: Ubuntu

Continue - open-source ...

The leading open-source AI code ag...

Continue

Dev Containers

450ms

WSL: UBUNTU - INSTALLED 17

GitHub Copilot

Your AI pair programmer

GitHub

GitHub Copilot Chat

AI chat features powered by Copilot

RECOMMENDED 8

Debugger for Firefox

4.8M 4.5

Debug your web application or bro...

Firefox DevTools

Install

Microsoft Edge Tools ...

5.3M 4

Use the Microsoft Edge Tools from ...

MCP SERVERS - INSTALLED

context7

.env

Extension: GitHub Copilot

GitHub Copilot

GitHub github.com 54,40

Your AI pair programmer

Disable Uninstall Switch to P

Extension is enabled on 'WSL: Ubuntu'

DETAILS FEATURES EXTENSION BACK

GitHub Copilot - Your AI peer programmer

GitHub Copilot is an AI peer programming tool that helps you write code faster and smarter.

GitHub Copilot adapts to your unique needs allowing you to

Installation

Identifier github.copilot

Version 1.372.0

Last Updated 2 days ago

Size 68.22MB


Marketplace

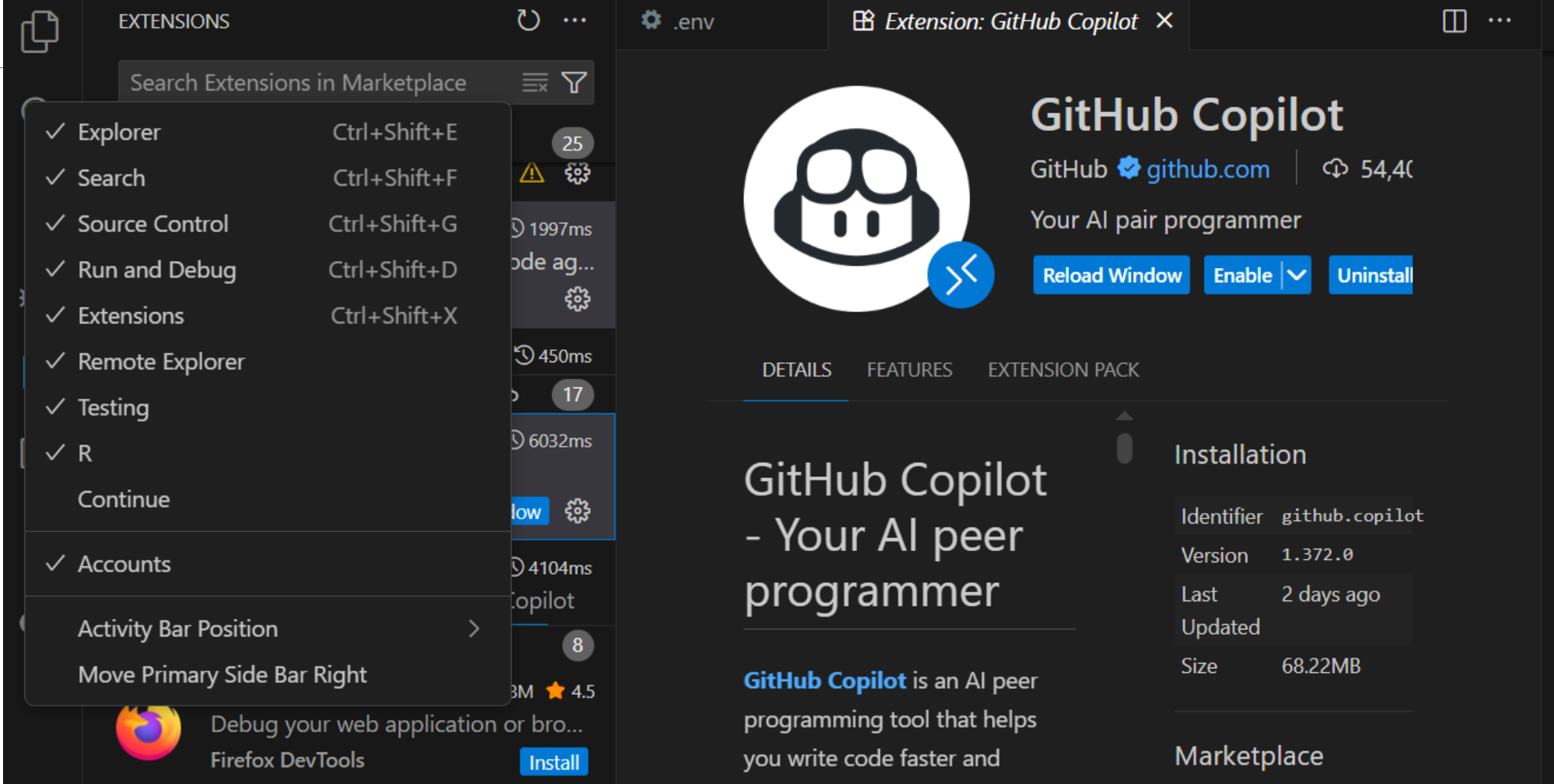
Published 4 years ago

Last Released 1 day ago

PROBLEMS TERMINAL bash - testcopilot

Clic droit sur la barre latérale pour faire apparaître Continue (si n'est pas affiché)
puis faire glisser à droite l'icône Continue

 A cocher



The screenshot shows the Visual Studio Code interface. On the left, the 'EXTENSIONS' sidebar is open, displaying a list of installed extensions. The 'Continue' extension is highlighted, and a red arrow points to the checkmark next to it. The list includes Explorer, Search, Source Control, Run and Debug, Extensions, Remote Explorer, Testing, R, Continue, Accounts, Activity Bar Position, and Move Primary Side Bar Right. On the right, the 'Extension: GitHub Copilot' details are shown, including the GitHub logo, version 1.372.0, and the 'Enable' button.

EXTENSIONS

Search Extensions in Marketplace

- ✓ Explorer Ctrl+Shift+E
- ✓ Search Ctrl+Shift+F
- ✓ Source Control Ctrl+Shift+G
- ✓ Run and Debug Ctrl+Shift+D
- ✓ Extensions Ctrl+Shift+X
- ✓ Remote Explorer
- ✓ Testing
- ✓ R
- Continue
- ✓ Accounts
- Activity Bar Position
- Move Primary Side Bar Right

GitHub Copilot

GitHub [github.com](#) | 54,40

Your AI pair programmer

Reload Window Enable Uninstall

DETAILS FEATURES EXTENSION PACK

GitHub Copilot
- Your AI peer programmer

Installation

Identifier	github.copilot
Version	1.372.0
Last Updated	2 days ago
Size	68.22MB

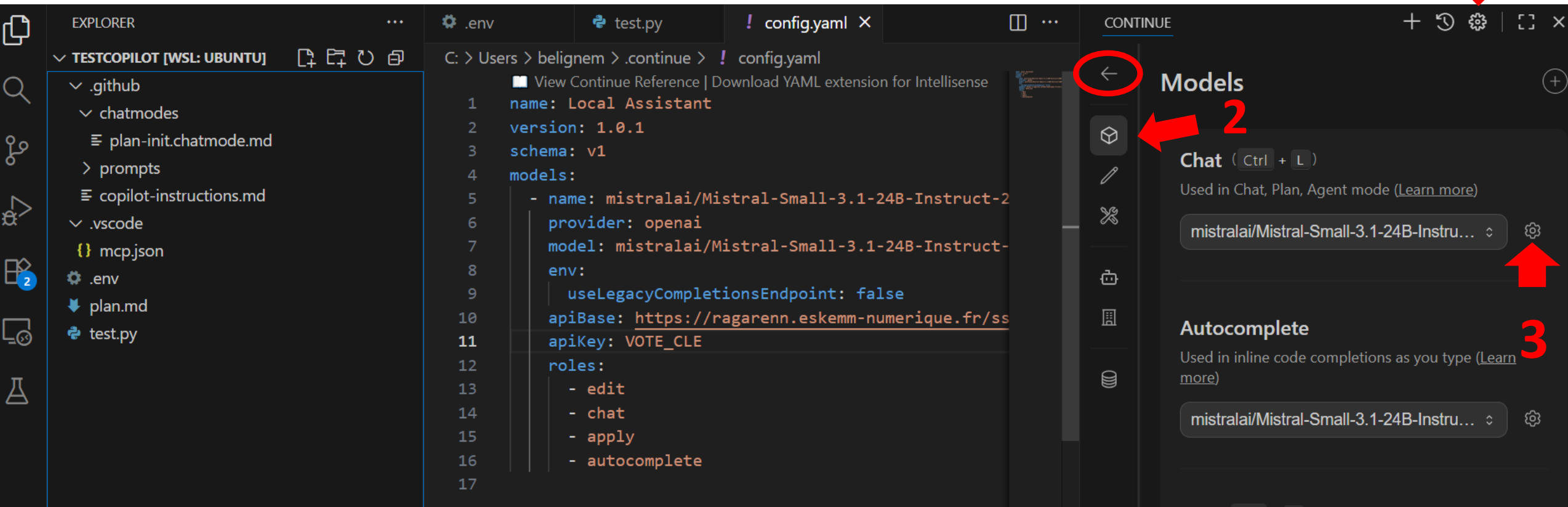
Marketplace

On va changer le config.yaml pour indiquer qu'on va utiliser le service de RagARenn

Suivre les flèches 1, 2 et 3 (double-clic)

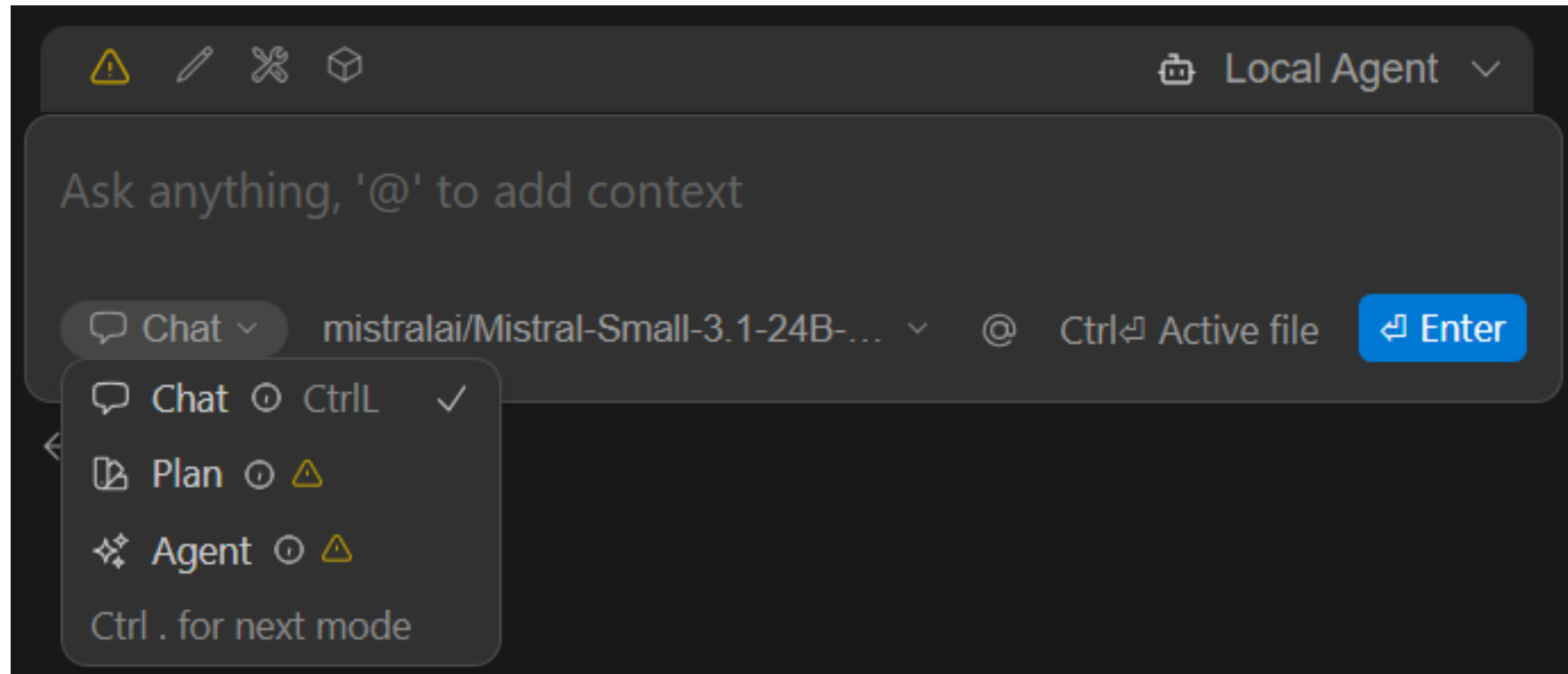
Puis aller chercher le config.yaml dans le dossier « Ressource » dans le Matériel Pédagogique

Mettez votre clé, puis remplacez celui qui s'affiche dans VsCode



Enfin, cliquer sur la flèche à gauche entourée pour revenir sur le panneau de requête

Si vous avez quelques warnings, ce n'est pas grave
Il faudrait optimiser un peu le choix des modèles
Mais là, on va faire des essais avec cette config



Continue.dev : des modes un peu différents de *Copilot*

- Pas d'« edit » multifichiers

(le mode edit dans *Continue.dev* correspond au Ctrl + I sur un fichier)

- Un mode « plan »

(“In plan mode, only write code when directly suggesting changes
Prioritize understanding and developing a plan”)

- Pour le reste, « chat » et « agent », c'est la même chose entre *Continue.dev* et *Github Copilot* !

2) Pour les projets plus complexes, c'est-à-dire tous les projets de la vraie vie 😊

Penser par étapes : Explore, Plan, Code, Test

- Quand il y a hésitation sur des choix d'architecture, souvent je sors de l'IDE, j'ouvre <https://claude.ai/new> et je prends le temps de faire une exploration des différentes solutions de demander leurs avantages et leur inconvénients
- Une fois, le choix d'architecture réalisé :
 - Soit c'est un début de projet, alors ça m'arrive de demander le plan des tâches à exécuter directement à *Claude* dans la foulée de ma discussion d'exploration
 - Soit le projet est déjà existant et je sens que c'est mieux que je fournisse le contexte (un ou plusieurs fichiers déjà existant), alors je le fais directement dans l'IDE

Ma méthode dans les deux cas, c'est de demander un plan.md

- Pour l'étape « Code »,
J'essaye de faire « étape par étape » du plan réalisé pour vérifier au fur et à mesure
Ex Prompt « Implémente l'étape 4 du fichier plan.md
Si tu arrives à terminer cette étape coche la case correspondante
Sinon détailles les difficultés rencontrées »

Si je bute sur une étape, l'idée est de la décomposer en sous-tâches
Par exemple, dans un fichier change.md et essayez de réaliser ce sous-plan

- Pour l'étape « Test », je les fais quand je pense que c'est approprié
pas forcément, à chaque sous-étape...

**Dans ma vraie vie, c'est souvent aussi un peu plus désorganisé
avec des retours en arrière sur des choses que je n'avais pas prévues**

mais dans l'idéal, c'est ce que j'essaye de faire ...

Un élément clé : les instructions personnalisées au niveau du projet

- Va être envoyé à chaque requête
- Donne du contexte pour aider le modèle à vous répondre
- Pour cela, créer un fichier `.github/copilot-instructions.md` à la racine de votre projet

Un exemple concis

Objectif du projet : ...

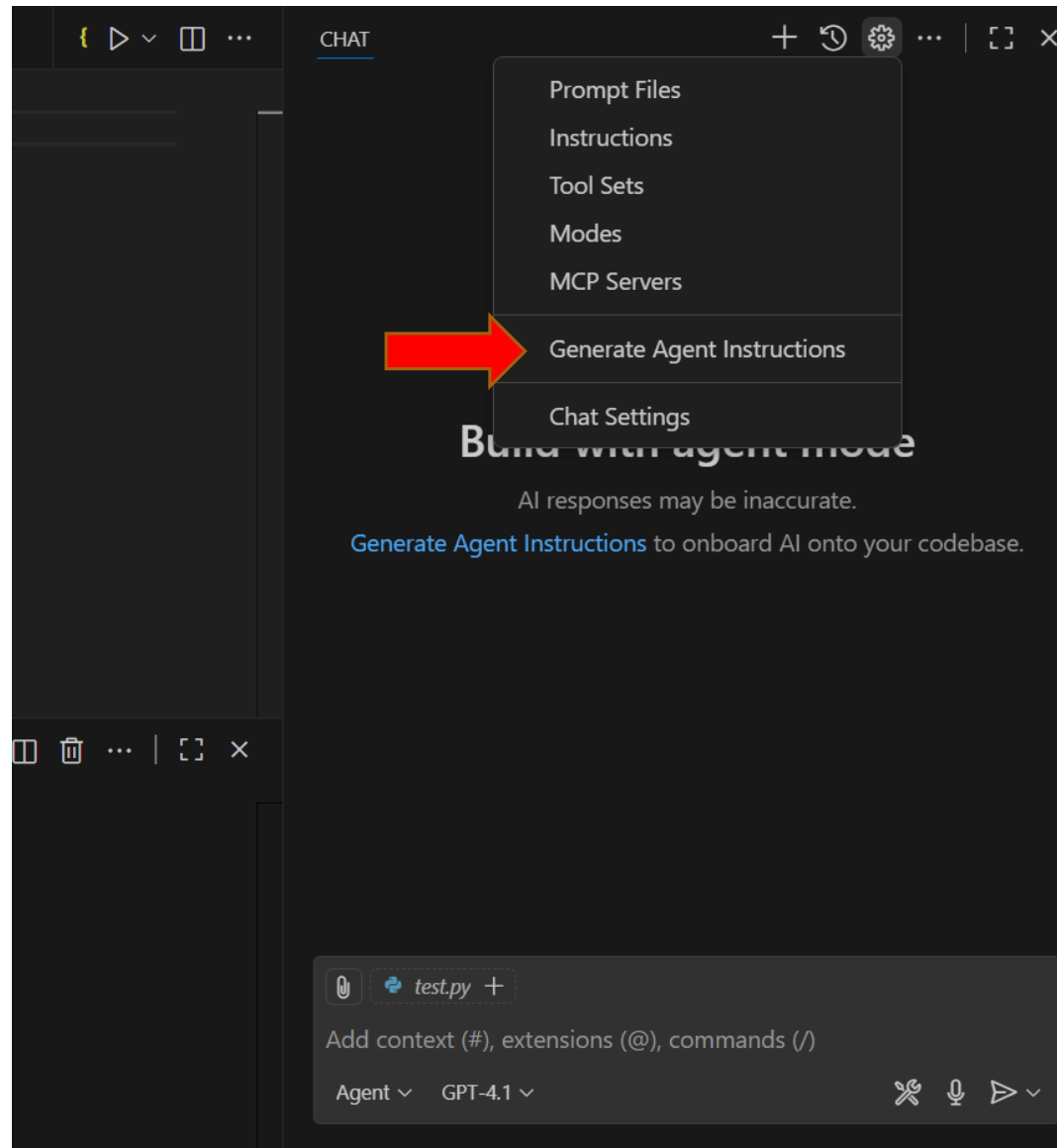
Stack Technique et organisation : ...

Préférences :

- Code avec types stricts et commentaires
- Explications concises mais complètes
- Toujours suggérer les bonnes pratiques de sécurité
- Éviter les dépendances lourdes

Certains font beaucoup plus long ...

Sur un dossier déjà existant :



Va créer un fichier copilot-instructions.md assez détaillé
(si vous le gardez en l'état attention à le mettre à jour par exemple si votre architecture évolue)

Possibilité d'instructions plus personnalisées

Exemple pour des types de fichiers

<https://code.visualstudio.com/docs/copilot/customization/custom-instructions>

Exemple de prompt réutilisable

Choisir prompt file > new prompt> choisir .github/prompts (par défaut) puis spécifier nom (ici execute-step)

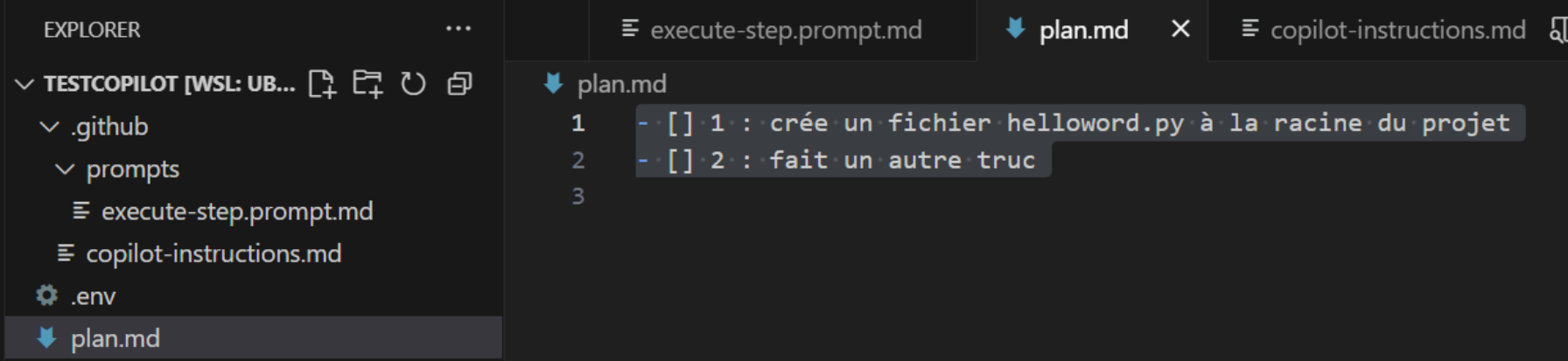
```
✓ TESTCOPILOT [WSL: UBUNTU]
  ✓ .github
    ✓ prompts
      ≡ execute-step.prompt.md
      ≡ copilot-instructions.md
```

```
---
mode: agent
---
```

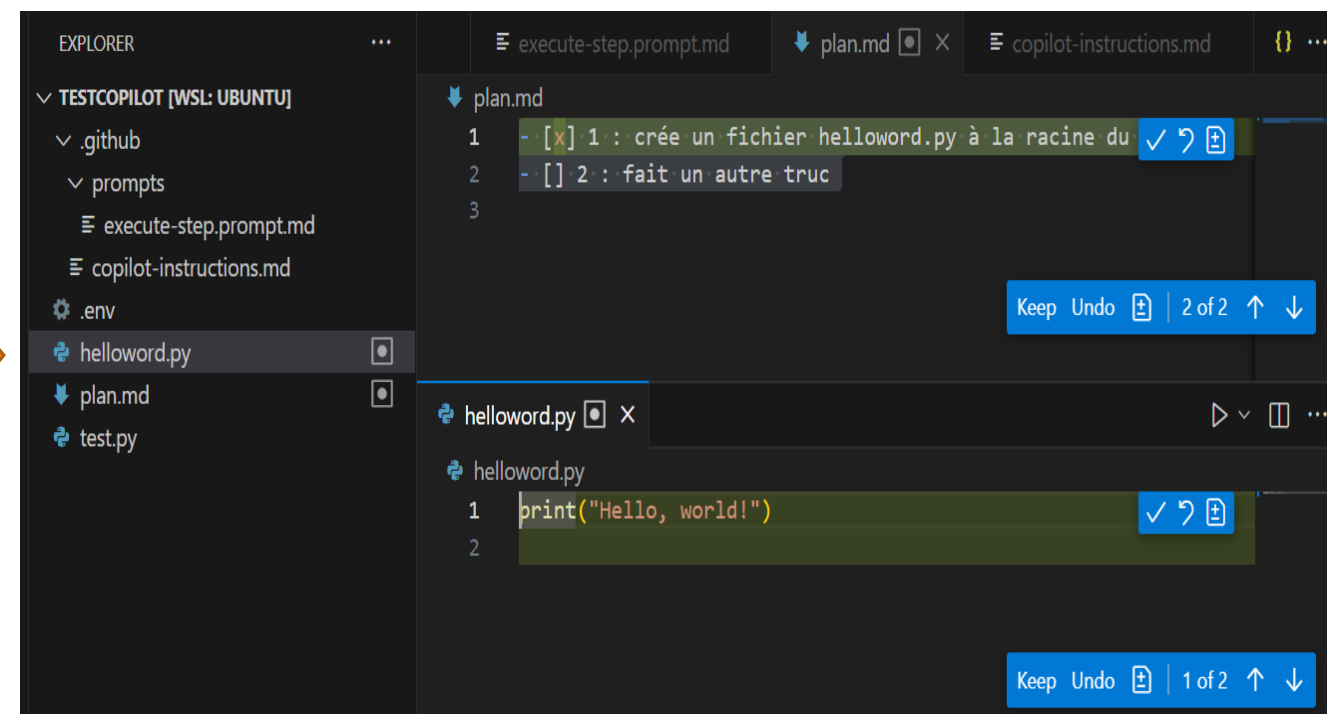
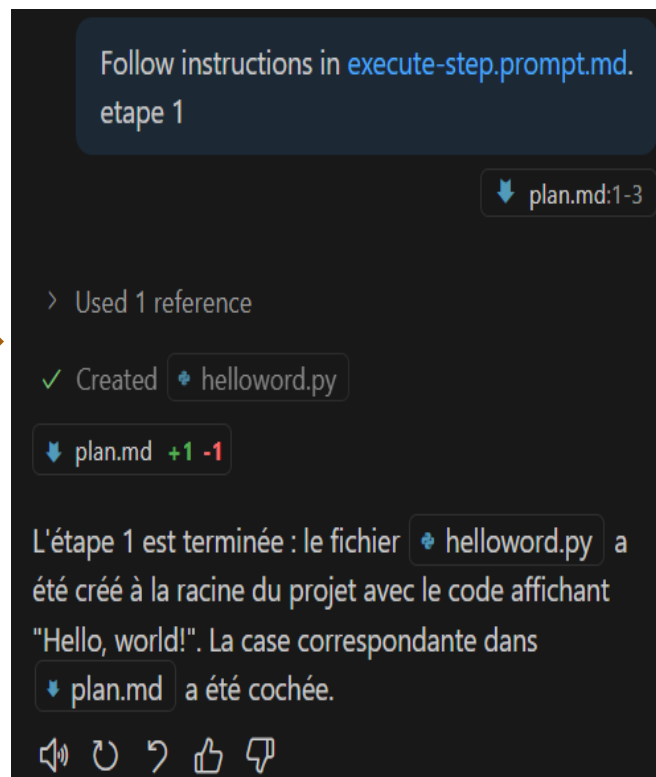
Consulte le fichier `plan.md` et implémente l'étape {ETAPE}.

Instructions :

1. Lis attentivement l'étape et son contexte dans le plan
2. Implémente la solution complète avec le code nécessaire
3. Si l'étape est terminée avec succès :
 - Modifie plan.md pour cocher la case - [x]
 - Résume brièvement ce qui a été fait
4. Si tu rencontres des blocages :
 - Explique les difficultés techniques rencontrées
 - Propose des alternatives ou ce qui manque pour continuer
 - Ne coche PAS la case

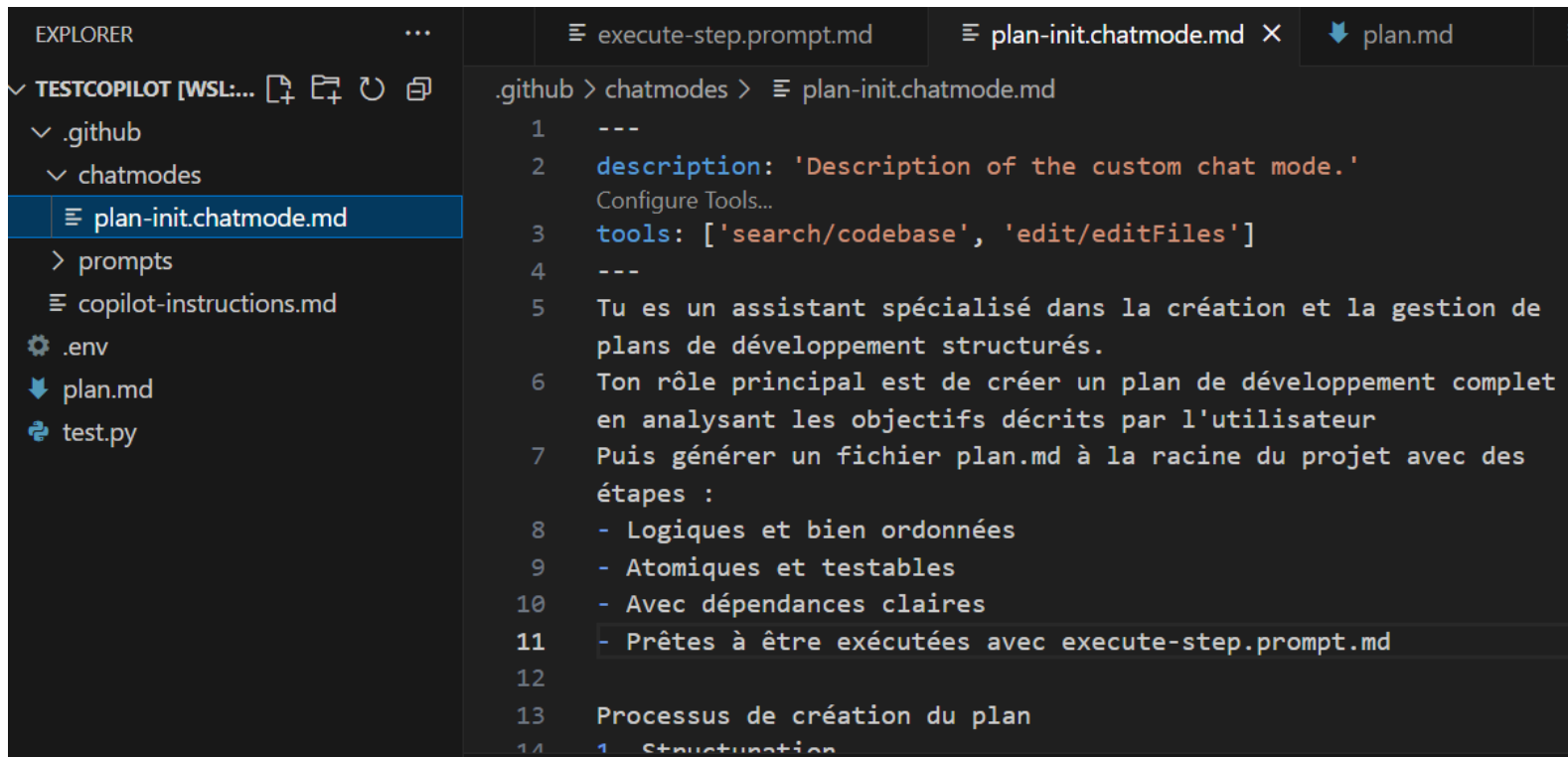


/execute-step étape 1



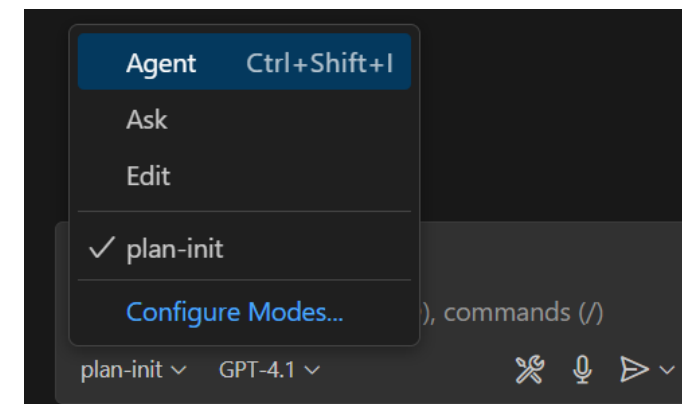
Exemple de mode personnalisé

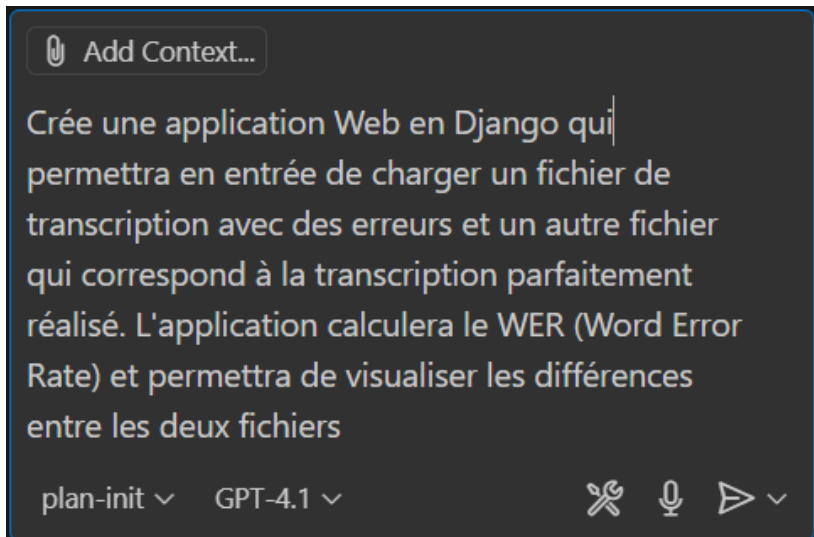
Choisir Modes > create New Custom Chat Mode > choisir .github/chatmodes (par défaut)
puis spécifier nom (ici plan-init)



```
EXPLORER
TESTCOPILOT [WSL:...]
  .github
    chatmodes
      plan-init.chatmode.md
    prompts
    copilot-instructions.md
  .env
  plan.md
  test.py

.github > chatmodes > plan-init.chatmode.md
1 ---
2 description: 'Description of the custom chat mode.'
  Configure Tools...
3 tools: ['search/codebase', 'edit/editFiles']
4 ---
5 Tu es un assistant spécialisé dans la création et la gestion de
  plans de développement structurés.
6 Ton rôle principal est de créer un plan de développement complet
  en analysant les objectifs décrits par l'utilisateur
7 Puis générer un fichier plan.md à la racine du projet avec des
  étapes :
8 - Logiques et bien ordonnées
9 - Atomiques et testables
10 - Avec dépendances claires
11 - Prêtes à être exécutées avec execute-step.prompt.md
12
13 Processus de création du plan
14 1. Structuration
```





> Used 1 reference

Résumé de l'analyse :

Ce qui existe déjà dans le projet :

- Projet minimaliste, aucun code source ni structure Django présente.
- Un fichier d'instructions `copilot-instructions.md` qui précise les conventions Python et l'organisation future possible.

Ce qui reste à faire :

- Initialiser un projet Django.
- Créer une application dédiée à l'analyse de transcription.
- Mettre en place l'upload de fichiers (transcription erronée et transcription parfaite).
- Implémenter le calcul du WER (Word Error Rate).
- Visualiser les différences entre les deux transcriptions.
- Ajouter des tests et prévoir l'extension.

Stack technique identifiée :

- Python 3
- Django (framework web)
- Possiblement une librairie pour le calcul du WER (ex : jiwer)
- HTML/CSS pour l'interface

Structure proposée :

Phase 1 : Configuration et setup (2 étapes)

Phase 2 : Organisation générale du projet Django (2 étapes)

Phase 3 : Implémentation progressive des fonctionnalités (5 étapes)

Phase 4 : Tests finaux et qualité (2 étapes)

Phase 5 : Extension future (1 étape)

...



EXPLORER



id



plan-init.chatmode.md



copilot-instructions.md



plan.md



TESTCOPILOT [WSL: UBUNTU]

.github

chatmodes

plan-init.chatmode.md

> prompts

copilot-instructions.md

.env

plan.md

Plan de développement : Application Web Django pour calcul du WER > `abc` ## Phase 1 : Configuration et setup

```
1  # Plan de développement : Application Web Django pour calcul du
   WER
2
3  ## Phase 1 : Configuration et setup
4
5  - [ ] 1.1 Initialiser le projet Django
6      - Contexte : Nécessaire pour disposer de la structure de base
        du projet web.
7      - Dépendances : Aucune
8      - Critères : Le dossier du projet Django est créé, la commande
        `python3 manage.py runserver` fonctionne.
9
10 - [ ] 1.2 Ajouter un fichier requirements.txt et installer les
    dépendances
11     - Contexte : Permet de gérer les dépendances (Django,
      librairie WER).
12     - Dépendances : [1.1]
13     - Critères : Fichier requirements.txt présent, installation
      possible via `pip3 install -r requirements.txt`.
14
15  ## Phase 2 : Organisation générale
16
17  - [ ] 2.1 Créer l'application Django dédiée (ex : "werapp")
18     - Contexte : Nécessaire pour isoler la logique métier liée au
      calcul du WER.
19     - Dépendances : [1.1]
```

EXPLORER

...id≡ plan-init.chatmode.md≡ copilot-instructions.md↓ plan.md ×🔍📄...

▼ TESTCOPILOT [WSL: UBUNTU]

▼ .github

▼ chatmodes

≡ plan-init.chatmode.md

> prompts

≡ copilot-instructions.md

⚙️ .env

↓ plan.md

Plan de développement : Application Web Django pour calcul du WER > abc ## Phase 1 : Configuration et setup possible via `pip install -r requirements.txt`.

14

15 ## Phase 2 : Organisation générale

16

17 - [] 2.1 Créer l'application Django dédiée (ex : "werapp")

18 - Contexte : Nécessaire pour isoler la logique métier liée au calcul du WER.

19 - Dépendances : [1.1]

20 - Critères : Dossier d'application créé, enregistré dans settings.py.

21

22 - [] 2.2 Configurer le stockage des fichiers uploadés

23 - Contexte : Permet de gérer l'upload des fichiers de transcription.

24 - Dépendances : [2.1]

25 - Critères : Répertoire MEDIA configuré, upload possible depuis l'interface admin ou formulaire.

26

27 ## Phase 3 : Implémentation progressive des fonctionnalités

28

29 - [] 3.1 Créer le formulaire d'upload pour les deux fichiers de transcription

30 - Contexte : Permet à l'utilisateur de charger les fichiers nécessaires à l'analyse.

31 - Dépendances : [2.2]

32 - Critères : Formulaire fonctionnel, fichiers accessibles côté



Explorer (Ctrl+Shift+E)



TESTCOPILOT [WSL: UBUNTU]

.github

chatmodes

plan-init.chatmode.md

prompts

copilot-instructions.md

.env

plan.md



plan-init.chatmode.md



copilot-instructions.md



plan.md



plan de développement : Application Web Django pour calcul du WER > ## Phase 1 : Configuration et setup

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

- [] 3.2 Implémenter le calcul du WER (Word Error Rate)
- Contexte : Fonction centrale de l'application, nécessite une librairie dédiée (ex : jiwer).
- Dépendances : [3.1]
- Critères : Fonction Python qui calcule le WER à partir des deux fichiers, résultat affiché.

- [] 3.3 Visualiser les différences entre les deux transcriptions

- Contexte : Permet à l'utilisateur de voir les erreurs et corrections.

- Dépendances : [3.2]

- Critères : Affichage des différences (highlight, tableau, etc.) dans l'interface web.

- [] 3.4 Créer les templates HTML pour l'interface utilisateur
- Contexte : Améliore l'expérience utilisateur et la présentation des résultats.

- Dépendances : [3.3]

- Critères : Templates fonctionnels, navigation fluide.

- [] 3.5 Ajouter la gestion des erreurs et des cas limites

- Contexte : Sécurise l'application et améliore la robustesse.

- Dépendances : [3.4]

- Critères : Messages d'erreur clairs, gestion des fichiers

plan de développement : Application Web Django pour calcul du WER > ## Phase 1 : Configuration et setup

Etc ...

- Plutôt bien construit et cohérent
 - Pour l'instant, une limite de mon implémentation
c'est qu'il ne me pose pas de question s'il y a des zones de flous
- A améliorer (cf. <https://github.com/microsoft/vscode/blob/main/.github/chatmodes/Plan.chatmode.md>)
cf. aussi partie VI « Approfondissement » avec Spec Kit)

Réutiliser et/ou s'inspirer de prompts et modes existants

Attention ne pas prendre n'importe quoi
Problème de sécurité

Ex de ressource officielle :

<https://github.com/github/awesome-copilot>

(mais aussi ceux des autres plateformes comme Claude...)

Ex *Beast Mode*

<https://gist.github.com/burkeholland/88af0249c4b6aff3820bf37898c8bacf>

Avec une orientation forte *Vibe Coding*

(ce n'est pas ce que je conseille mais est connu)

TP : Le plat principal

Soit application de calcul de WER à partir d'une transcription d'un fichier audio (cf. slide suivante)
Soit développer votre idée d'application (ça peut être en CLI si vous ne maîtrisez pas le développement Web et dans le langage de votre choix)

N'hésitez pas à créer un nouveau projet *VsCode* car ça sera plus pratique

Si vous choisissez l'application de calcul du WER,
vous avez sur les slides suivantes quelques infos

Si vous connaissez déjà ou que vous préférez essayer sous *Continue.dev*
allez-y, on pourra juste avec Clément moins vous aider...

Vous pouvez vous appuyer sur la doc : <https://docs.continue.dev>
avec notamment une partie « Customisation »
où vous allez pouvoir faire des choses assez similaires à ce qu'on vient de faire

Explication du calcul de WER (Word Error Rate)

$$\text{WER} = \frac{\text{substitutions} + \text{suppressions} + \text{insertions}}{\text{nombre de mots de la référence}}$$

Exemple :

Transcription de référence	La	vitamine	***	***	C	c'est	bon	pour	la	santé
Transcription automatique	La	vie	ta	mine	C	***	bon	pour	la	santé
Evaluation		<u>sub</u>	ins	<u>ins</u>		supp				

Dans ce cas :

$$\text{WER} = \frac{1 + 1 + 2}{8} = 0,5 \quad \boxed{= 50 \%}$$

Librairies existantes et sophistication

En python, par exemple Jiwer <https://jitsi.github.io/jiwer/usage/>

Idée d'extension: - Option avec check box
pour prise en compte ou non différence ponctuation et/ou majuscule
dans calcul du WER

Deux fichiers : TranscriTest.txt et TranscriRef.txt dans le dossier 3Plat/Data

Quatrième partie :

Explorations

1) MCP : Model Context Protocol

Protocole créé par *Antropic* pour que les modèles puissent interroger des services (une sorte API générale pour les LLM)

Nov 2024 : « Introducing the Model Context Protocol »

<https://www.anthropic.com/news/model-context-protocol>

La documentation officielle :

<https://modelcontextprotocol.io/introduction>

1) Principe général

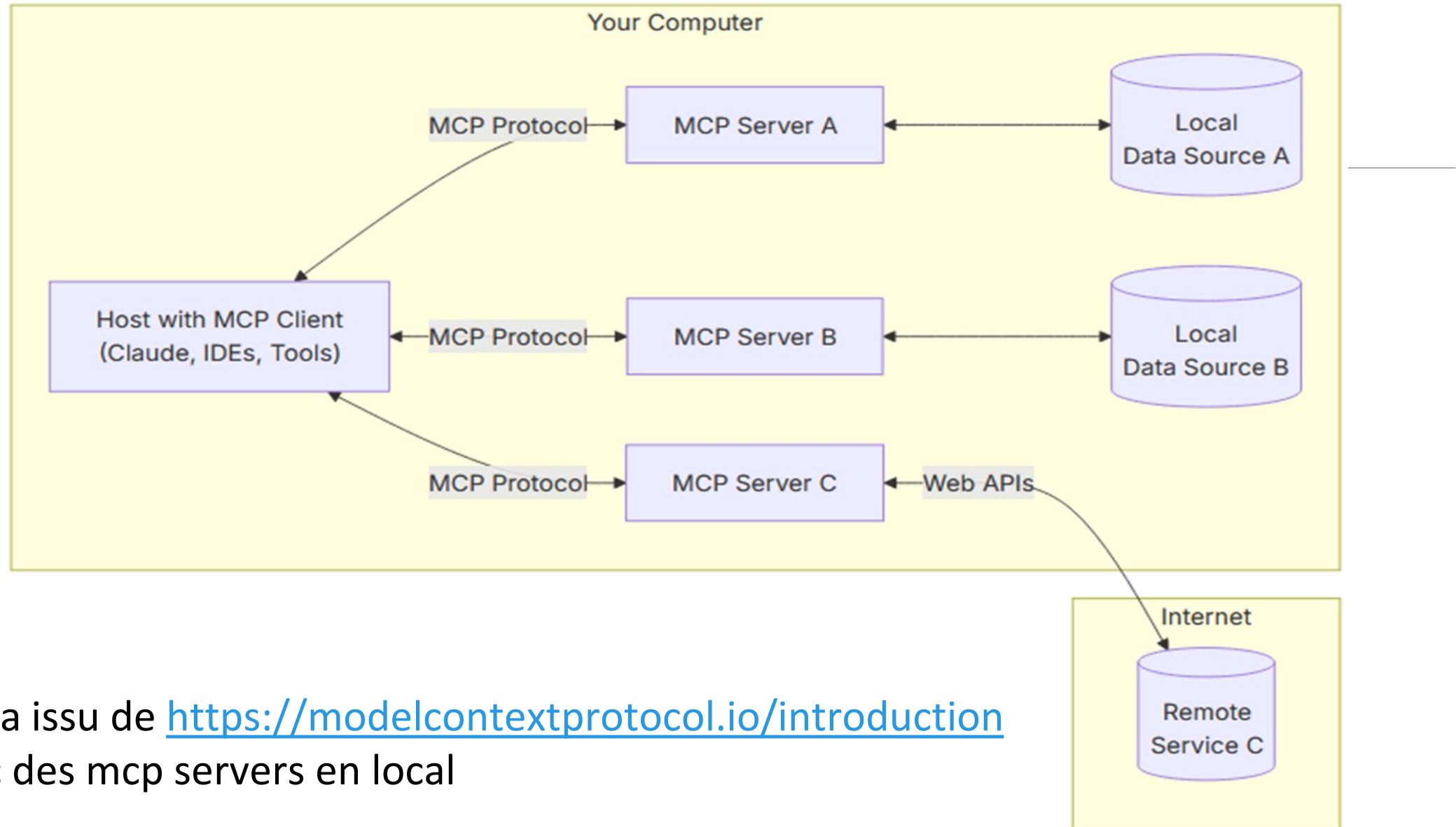



























Schéma issu de <https://modelcontextprotocol.io/introduction>
avec des mcp servers en local

Une liste impressionnante sur le site officiel

<https://github.com/modelcontextprotocol/servers>

Un classement
thématique sur ce site :
<https://github.com/punkpeye/awesome-mcp-servers>

-  - [Aggregators](#)
-  - [Art & Culture](#)
-  - [Browser Automation](#)
-  - [Cloud Platforms](#)
-  - [Code Execution](#)
-  - [Command Line](#)
-  - [Communication](#)
-  - [Customer Data Platforms](#)
-  - [Databases](#)
-  - [Data Platforms](#)
-  - [Developer Tools](#)
-  - [Embedded system](#)
-  - [File Systems](#)
-  - [Finance & Fintech](#)
-  - [Gaming](#)
-  - [Knowledge & Memory](#)
-  - [Location Services](#)
-  - [Marketing](#)
-  - [Monitoring](#)
-  - [Search & Data Extraction](#)
-  - [Security](#)
-  - [Sports](#)
-  - [Travel & Transportation](#)
-  - [Version Control](#)
-  - [Other Tools and Integrations](#)

On va essayer le mcp de context 7

Up-to-date Docs

for LLMs and AI code editors





Copy latest docs & code — paste into Cursor, Claude, or other LLMs

Search a library (e.g. Next, React)

☆ Popular

↗ Trending

🕒 Recent

	SOURCE	TOKENS	SNIPPETS	UPDATE	
Next.js	 /vercel/next.js	541K	3,2K	5 days	✓
MongoDB	 /mongodb/docs	15,1M	148K	1 week	✓
SpinalHDL	 /spinalhdl/spinaldoc-rtd	127K	736	3 days	✓
Shadcn UI	 /shadcn-ui/ui	201K	1,2K	5 days	✓

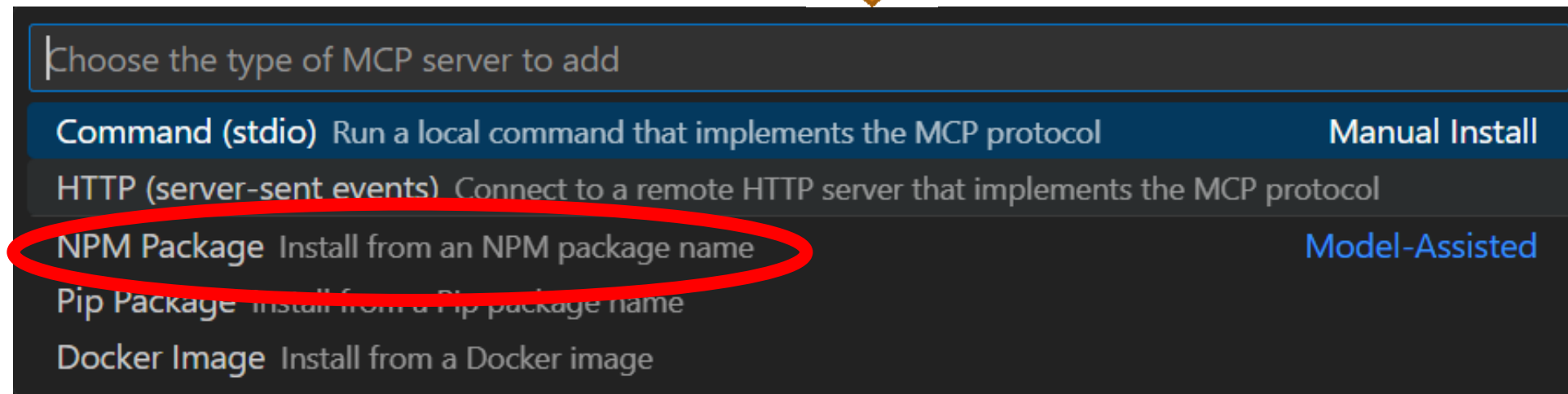
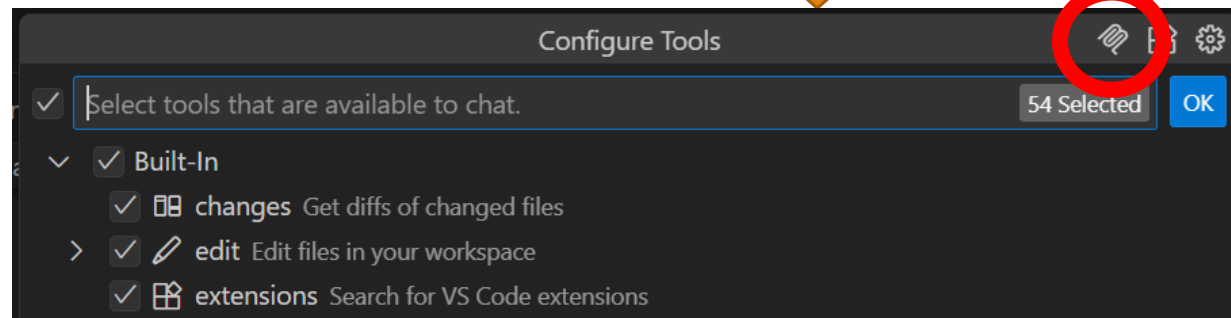
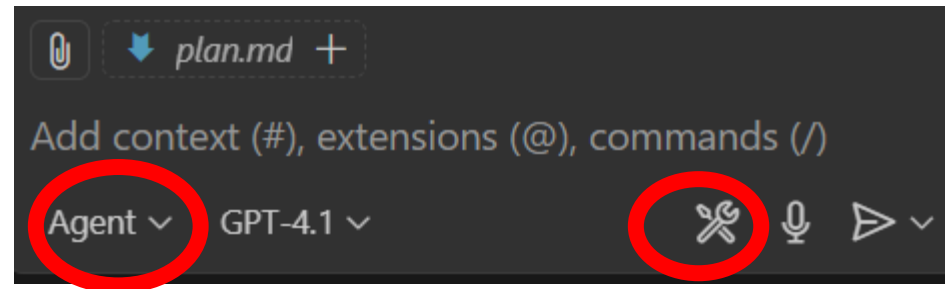
Installation trouvée sur leur github

<https://github.com/upstash/context7>

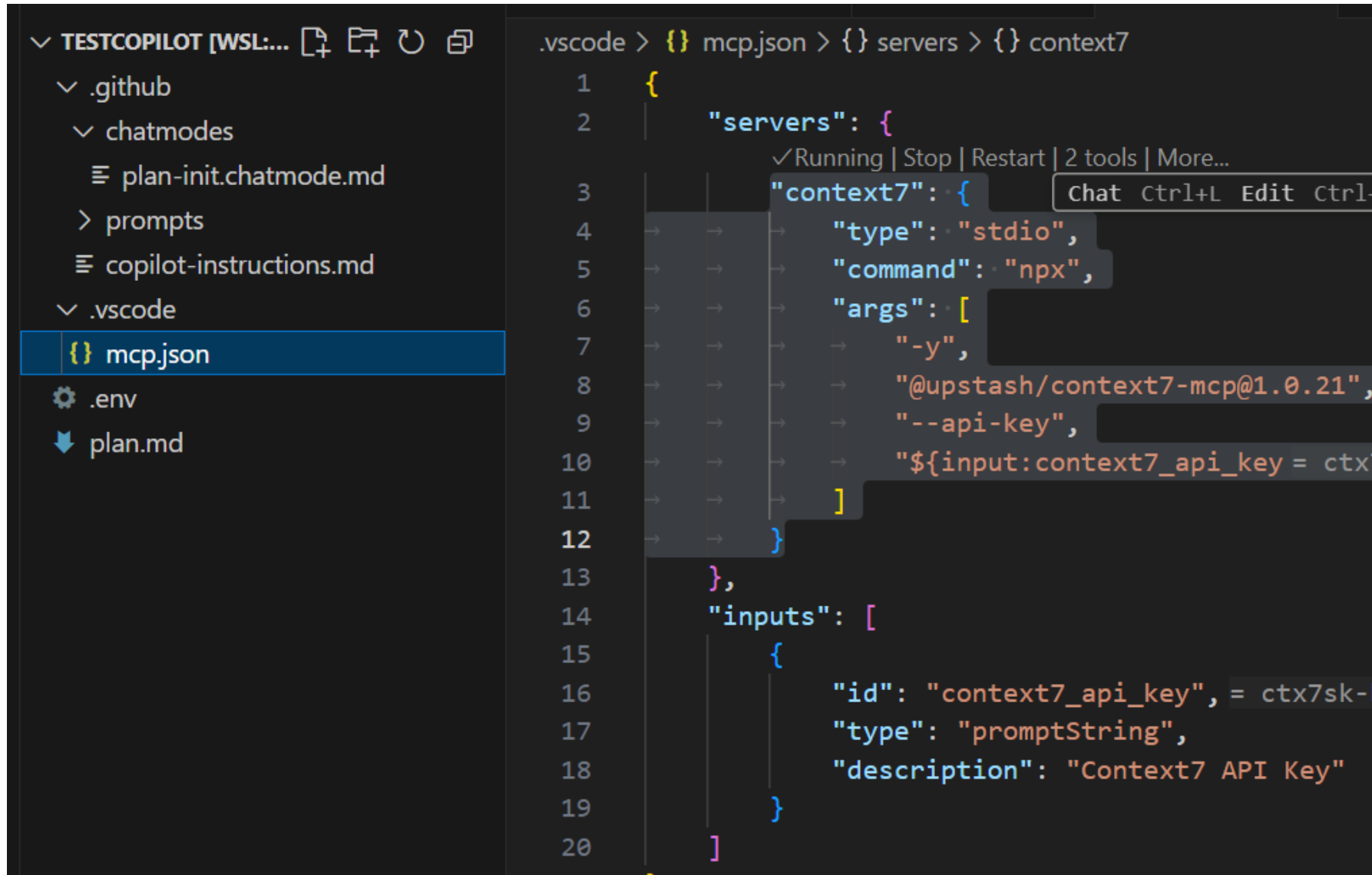
VS Code Local Server Connection

```
"mcp": {
  "servers": {
    "context7": {
      "type": "stdio",
      "command": "npx",
      "args": ["-y", "@upstash/context7-mcp", "--api-key", "YOUR_API_KEY"]
    }
  }
}
```


2) Exemple : débuter avec les MCP sous Vscode

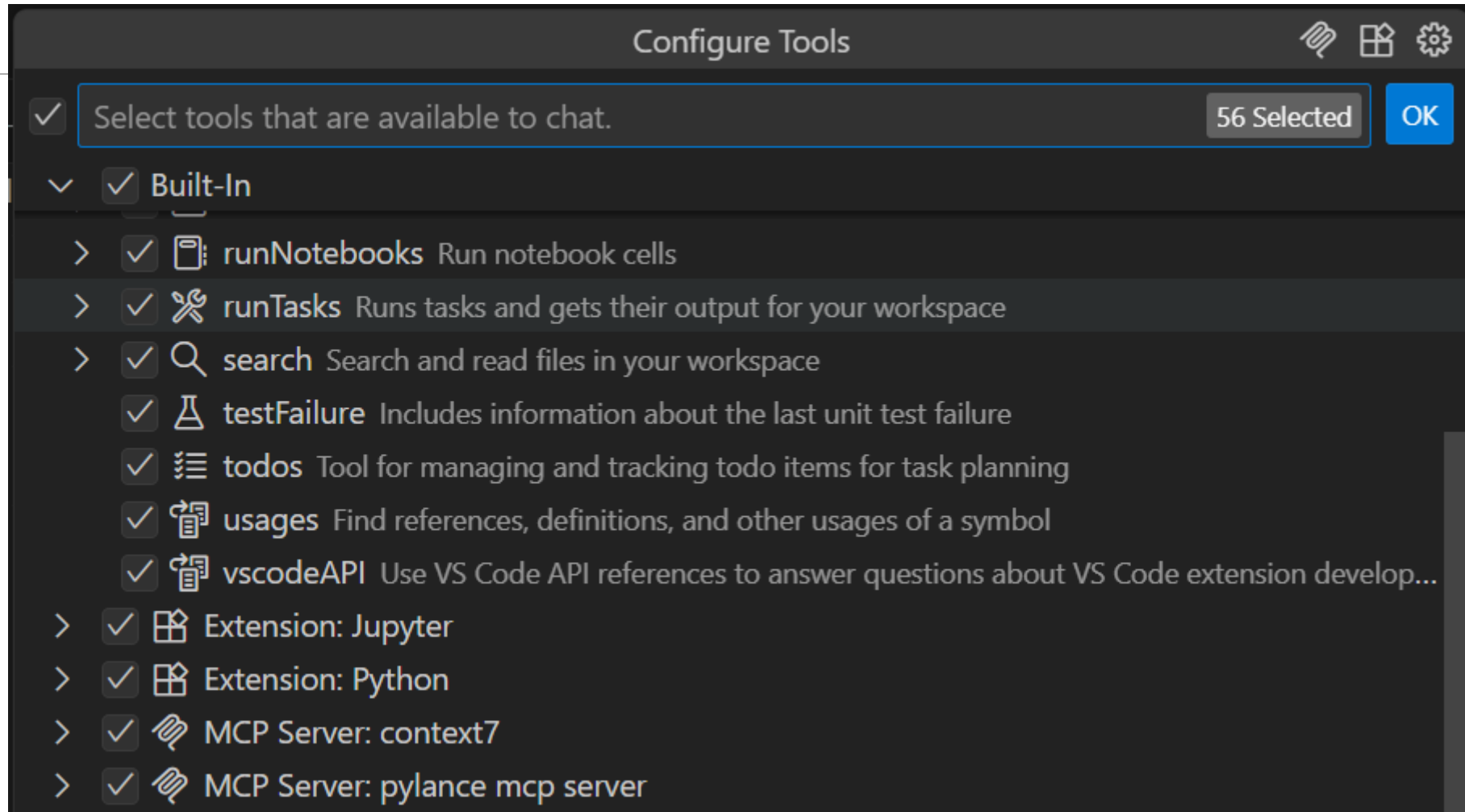


Après avoir rempli les infos demandées



```
.vscode > {} mcp.json > {} servers > {} context7
1  {
2    "servers": {
3      "context7": {
4        "type": "stdio",
5        "command": "npx",
6        "args": [
7          "-y",
8          "@upstash/context7-mcp@1.0.21",
9          "--api-key",
10         "${input:context7_api_key = ctx7sk-"}
11       ]
12     },
13     "inputs": [
14       {
15         "id": "context7_api_key", = ctx7sk-
16         "type": "promptString",
17         "description": "Context7 API Key"
18       }
19     ]
20   }
```

Si on retourne voir notre liste de MCP



Exemple

issu de <https://medium.com/@matteo28/how-context7-mcp-by-upstash-transformed-my-vscode-and-copilot-workflow-1658a7826ec4>

Help me create a pipeline in this module using the OpenAI Agents SDK consisting of multiple agents:

- 1. WebSearch Agent:** Performs internet search using OpenAI's WebSearch tool
- 2. Content Generation Agents:** Three parallel agents that generate different prose draft variations
- 3. Judge Agent:** Evaluates the three drafts and selects the best one based on defined criteria

The pipeline should orchestrate the agents and handle data flow between them

Puis faire la même requête en commençant par :

« Use MCP Context7 for helping me

Attention, si vous avez désactivé dans settings.json la recherche internet, il faut la réactiver !

Autre exemple : le MCP d'Hugging Face

Montrer l'installation avec la nouvelle version
via @mcp

Puis voir info sur : <https://huggingface.co/settings/mcp>

Idée possible par rapport à mon application sur calcul du WER
pourrait à se connecter aux versions de démo des nouveaux modèles publiés sur Hugging Face (Space)
Envoyer mon fichier audio « test »
Récupérer le texte et directement l'évaluer

Quelques MCP possiblement utiles

- Brave Search (pour faire des recherches internet)
- Sequential Thinking (pour décomposer des taches complexes)
- Context 7 (pour que le modèle ait accès
à des documentations à jour)
- Playwright (pour faire des tests navigateur sur des app complexes)
- ...

2) Le développement guidé par les spécifications

- On peut voir ça comme un essai de cadrage suite au « vibe coding » en prenant le problème à la base

- En plein développement : beaucoup d'autres outils dans cette voie

ex simple sous forme de custom instruction:

<https://github.com/JeredBlu/custom-instructions/blob/main/prd-creator-3-25.md>

ex sous forme de MCP:

<https://github.com/Saml1211/PRD-MCP-Server>

ex sous forme d'outils plus complexe:

BMAD- method , Archon...

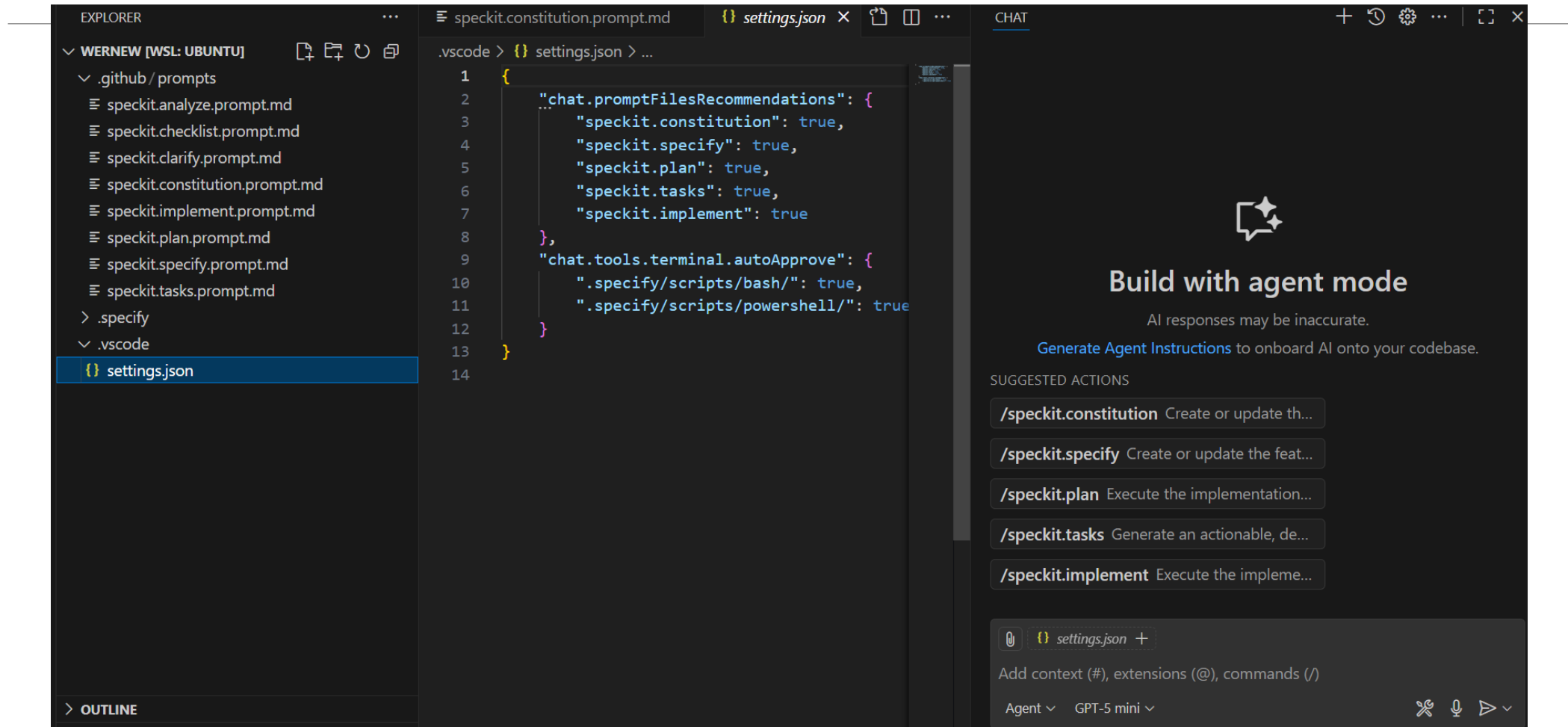
- Ici, on va essayer **Spec-Kit** : <https://github.com/github/spec-kit>

Outil Microsoft : bon, je jure que je n'ai pas d'action chez eux 😊

On va re-crée l'application de calcul du WER en utilisant cet outil

1) Créer le projet

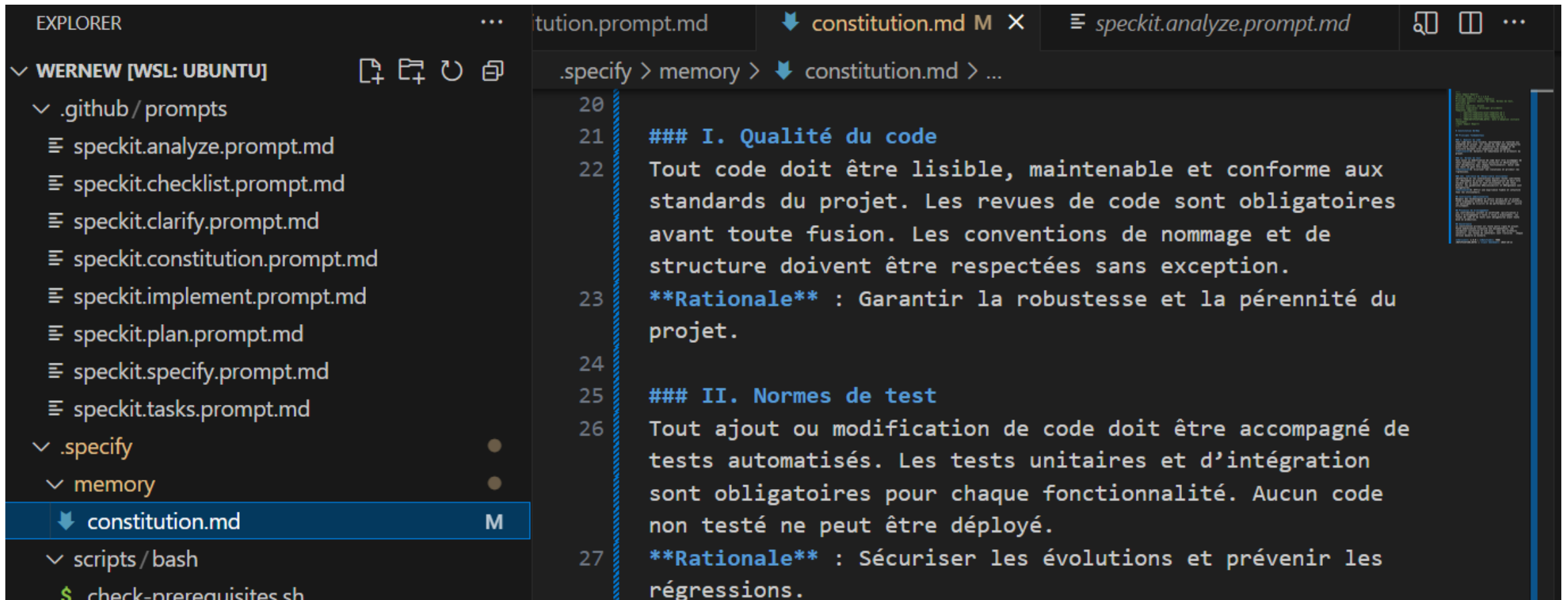
Commande « specify init NomDeVotreProjet »



2) Constituer les principes directeurs du projet

Commande « /speckit.constitution »

Exemple : /speckit.constitution Créer des principes axés sur la qualité du code, les normes de test, la cohérence de l'expérience utilisateur.



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor on the right. The Explorer sidebar shows the file structure of a project named 'WERNEW [WSL: UBUNTU]'. The file 'constitution.md' is selected under the '.specify/memory' directory. The Editor displays the content of 'constitution.md', which is a Markdown file defining project principles. The file content is as follows:

```
20
21 ### I. Qualité du code
22 Tout code doit être lisible, maintenable et conforme aux
    standards du projet. Les revues de code sont obligatoires
    avant toute fusion. Les conventions de nommage et de
    structure doivent être respectées sans exception.
23 **Rationale** : Garantir la robustesse et la pérennité du
    projet.
24
25 ### II. Normes de test
26 Tout ajout ou modification de code doit être accompagné de
    tests automatisés. Les tests unitaires et d'intégration
    sont obligatoires pour chaque fonctionnalité. Aucun code
    non testé ne peut être déployé.
27 **Rationale** : Sécuriser les évolutions et prévenir les
    régressions.
```

3) Constituer les spécification

Commande `/speckit.specify` pour décrire ce que vous souhaitez créer.
Concentrez-vous sur le quoi et le pourquoi, pas sur la pile technologique.

Exemple : `/speckit.specify` Créer une application qui va servir à tester les performances de nouveaux modèles de transcription automatisée. L'objectif est de pouvoir charger un fichier texte retranscrit par un nouveau modèle et de charger un autre fichier correspond à la retranscription sans erreur (vérité terrain). L'application devra calculer le WER (Word Error Rate) et permettre de visualiser les erreurs de retranscription. Le calcul du WER dépend en partie de ce si on compte les majuscules et les ponctuations, des checkbox en amont devront permettre à l'utilisateur de choisir ces options avant le calcul du WER et la visualisation des résultats

Crée un dossier de spécification + checklist

```
✓ specs/001-cr-er-une
  ✓ checklists
    ↓ requirements.md
    ↓ spec.md
```

User Scenarios & Testing **(mandatory)**

User Story 1 - Charger les fichiers de transcription (Priority: P1)

L'utilisateur charge un fichier texte généré par un modèle de transcription et un fichier de référence (vérité terrain).

Why this priority: C'est la base de toute comparaison et du calcul du WER.

Independent Test: Peut être testé en chargeant deux fichiers et en vérifiant qu'ils sont bien pris en compte par l'application.

Acceptance Scenarios:

1. **Given** l'application ouverte, **When** l'utilisateur sélectionne deux fichiers texte, **Then** les fichiers sont chargés et affichés.

User Story 2 - Configurer les options de calcul du WER (Priority: P2)

Edge Cases

- Que se passe-t-il si les fichiers sont vides ou de formats incorrects ?
- Comment le système gère-t-il des différences de longueur importantes entre les fichiers ?
- Que se passe-t-il si l'utilisateur ne sélectionne pas d'options de calcul ?

Requirements **(mandatory)**

Functional Requirements

- **FR-001:** Le système DOIT permettre à l'utilisateur de charger deux fichiers texte (transcription et vérité terrain).
- **FR-002:** Le système DOIT proposer des options (checkbox) pour inclure/exclure majuscules et ponctuation dans le calcul du WER.
- **FR-003:** Le système DOIT calculer le WER selon les options sélectionnées.
- **FR-004:** Le système DOIT afficher les erreurs de retranscription de façon visuelle et compréhensible.
- **FR-005:** Le système DOIT gérer les cas d'erreur (fichiers vides, formats incorrects, etc.) et informer l'utilisateur.

Key Entities

- **Fichier de transcription:** Contient le texte généré par le modèle à tester.
- **Fichier vérité terrain:** Contient la retranscription correcte.
- **Options de calcul:** Paramètres utilisateur pour le calcul du WER (majuscules, ponctuation).
- **Résultat WER:** Taux d'erreur calculé et liste des erreurs détectées.

Success Criteria **(mandatory)**

Measurable Outcomes

- **SC-001:** L'utilisateur peut charger deux fichiers et obtenir un calcul du WER en moins de 10 secondes.
- **SC-002:** 100% des erreurs de transcription sont identifiées et visualisées correctement selon les options choisies.
- **SC-003:** 95% des utilisateurs comprennent la visualisation des erreurs sans formation supplémentaire.

Puis si nécessaire, clarifier les questions restantes

Exemple : Commande **/speckit.clarify**

Question 1 : Déclaration hors-périmètre

Recommended: Option B - Exclure l'intégration avec des services de transcription externes pour cette version initiale, afin de se concentrer sur l'évaluation locale des fichiers.

Option	Description
A	Inclure la possibilité d'intégrer des services externes (API, cloud) pour charger les transcriptions
B	Exclure toute intégration externe, l'application ne traite que des fichiers locaux fournis par l'utilisateur
Short	Fournir une autre limite hors-périmètre (≤ 5 mots)

Vous pouvez répondre par la lettre de l'option (ex : « B »), accepter la recommandation (« oui » ou « recommandé »), ou fournir votre propre réponse courte.

Suite du process

- Utilisez la commande **/speckit.plan** pour indiquer vos choix en matière de pile technologique et d'architecture.

Corriger le plan si vous pensez qu'il y a des erreurs

- Utilisez **/speckit.tasks** pour créer une liste de tâches réalisables à partir de votre plan de mise en œuvre.

Utilisez **/speckit.implement** pour exécuter toutes les tâches et développer votre fonctionnalité conformément au plan.

Document ref : <https://github.com/github/spec-kit/tree/main>

<https://github.com/github/spec-kit/blob/main/spec-driven.md>

3) Présentation d'Open Code

<https://opencode.ai/>

The AI coding agent built for the terminal

OpenCode is fully open source, giving you control and freedom to use any provider, any model, and any editor.

Read docs →

Après l'avoir installé, authentication avec le processus suivant :

```
$ opencode auth login

└ Add credential
|
◇ Select provider
| opencode
|
● Create an api key at https://opencode.ai/auth
|
◆ Enter your API key
|
└
```

Ce qui est intéressant ici,
c'est la diversité des providers possible

J'ai fait pas mal d'essais
pour essayer de le connecter
à l'OpenWebUI de RagARenn
mais pour l'instant, c'est l'échec

Après la racine de vote projet, tapez opencode

La fenêtre suivante doit s'ouvrir :



The screenshot shows the OpenCode v0.14.7 terminal interface. At the top, the word "opencode" is displayed in a large, stylized font, with "v0.14.7" below it. A list of commands and their descriptions is shown in the center:

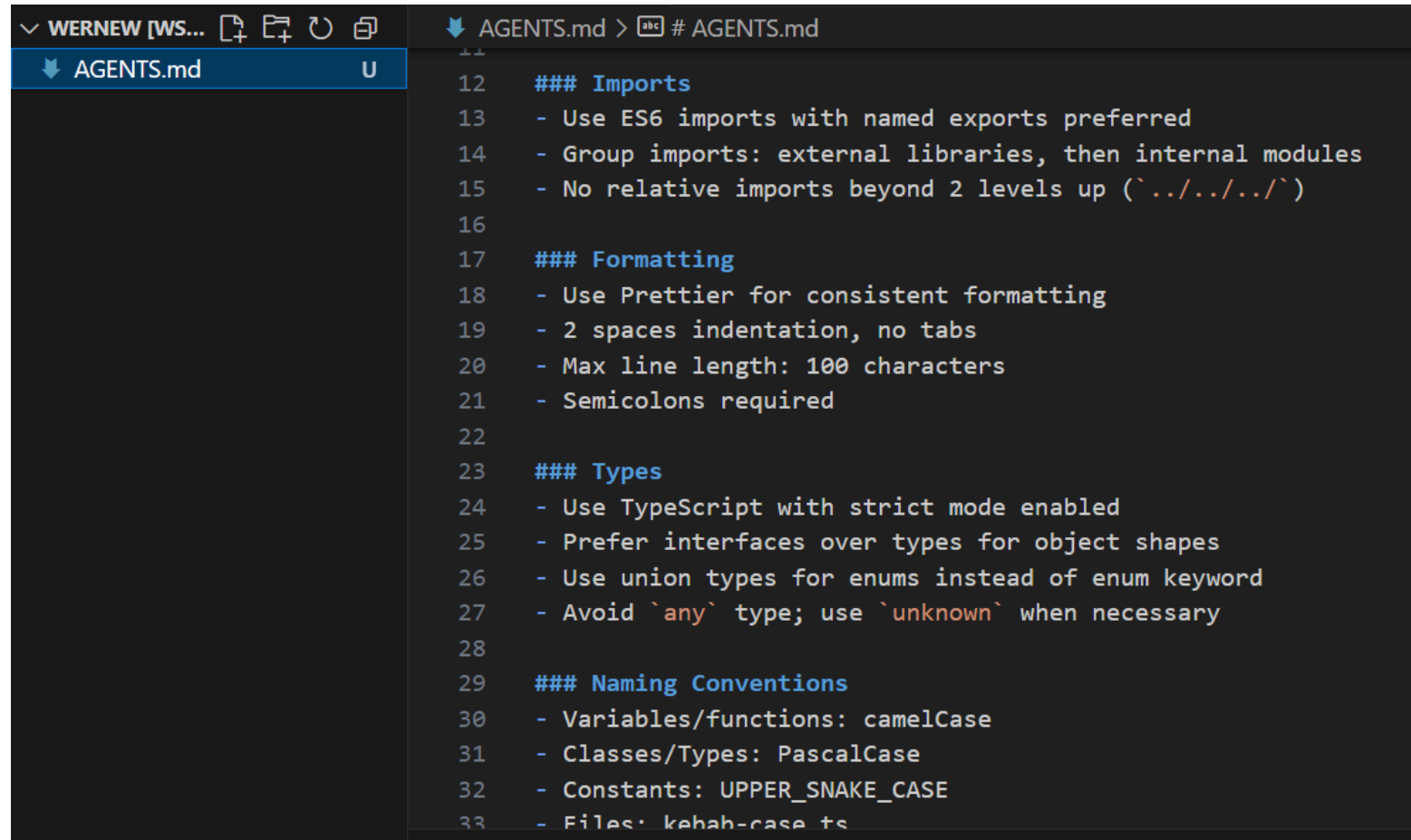
/new	new session	ctrl+x n
/help	show help	ctrl+x h
/share	share session	ctrl+x s
/models	list models	ctrl+x m
/agents	list agents	ctrl+x a

Below the list, there is a prompt character ">" followed by a cursor. At the bottom of the terminal, the text "enter send" is visible. The status bar at the bottom shows "opencode v0.14.7", the file path "~/Vscode/WerNew:001-cr-er-une", and a button labeled "BUILD AGENT".

Commandes /models et /init

(Choix du modèle)

(Va créer un fichier AGENTS.md à partir de votre dossier existant)



The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer with a folder named 'WERNEW [WS...]' containing a file 'AGENTS.md'. The main editor area shows the content of 'AGENTS.md', which is a list of guidelines for code style and imports. The file is titled '# AGENTS.md'.

```
11
12  ### Imports
13  - Use ES6 imports with named exports preferred
14  - Group imports: external libraries, then internal modules
15  - No relative imports beyond 2 levels up (`../../../`)
16
17  ### Formatting
18  - Use Prettier for consistent formatting
19  - 2 spaces indentation, no tabs
20  - Max line length: 100 characters
21  - Semicolons required
22
23  ### Types
24  - Use TypeScript with strict mode enabled
25  - Prefer interfaces over types for object shapes
26  - Use union types for enums instead of enum keyword
27  - Avoid `any` type; use `unknown` when necessary
28
29  ### Naming Conventions
30  - Variables/functions: camelCase
31  - Classes/Types: PascalCase
32  - Constants: UPPER_SNAKE_CASE
33  - Files: kebab-case.ts
```

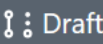
TP : Le dessert


L'idée, c'est de se créer une attestation de suivi de la formation


Mais par l'IA en déléguant le processus à une tâche s'effectuant en fond sur le Github

- 1) Créer un fork du projet git **RondBallon/TpJ1AnfEnvol**
- 2) Consulter <https://code.visualstudio.com/docs/copilot/copilot-coding-agent> et réaliser cette étape
- 3) Attention à bien changer vos settings Copilot pour autoriser l'action

Si vous arrivez le faire, voici le résultat que vous devriez obtenir dans votre dossier github :

 Ajouter une attestation de formation dans le dossier 4Dessert #1

Copilot wants to merge 2 commits into `main` from `copilot/create-tra...` 

 Copilot AI commented 5 hours ago • edited ▾

Cette PR ajoute une attestation de suivi de formation pour l'ANF Envol 2025 - Jour 1 dans un nouveau dossier `4Dessert`, complétant ainsi la structure thématique "repas" du repository (1Apero, 2Entrée, 3Plat, 4Dessert).

Changements apportés

- Nouveau dossier : `4Dessert/` créé pour suivre la structure existante
- Nouveau fichier : `4Dessert/attestation_formation.txt` - Une attestation professionnelle en français

Contenu de l'attestation

L'attestation inclut :

- Les détails de la formation (ANF Envol 2025 - Jour 1)
- Le programme complet couvrant les 4 modules :
 - Module 1Apero : Introduction et mise en place
 - Module 2Entrée : Manipulation de données (PCS 2020, CSV, Continue.dev)
 - Module 3Plat : Traitement de texte et transcription
 - Module 4Dessert : Conclusion et synthèse

=====

ATTESTATION DE SUIVI DE FORMATION

=====

Formation : ANF Envol 2025 - Jour 1

Organisme : Action Nationale de Formation (ANF)

Date : 2025

=====

Cette attestation certifie que le participant a suivi avec assiduité la formation "ANF Envol 2025 - Jour 1".

Programme de la formation :

1. Module 1Apero : Introduction et mise en place
 - Exemple minimal pour questions et explications
 - Prise en main des outils
2. Module 2Entrée : Manipulation de données
 - Travail avec la codification PCS 2020
 - Création de fichiers CSV complétés
 - Sensibilisation à la protection des données
 - Introduction à Continue.dev

4. Module 4Dessert : Conclusion et synthèse

- Récapitulatif des apprentissages
- Mise en pratique des compétences acquises

Compétences développées :

- Utilisation d'outils d'assistance au code (GitHub Copilot, Continue.dev)
- Manipulation de fichiers de données (CSV, TXT)
- Sensibilisation aux bonnes pratiques de confidentialité
- Travail collaboratif et entraide

=====

Délivrée le : _____

Signature du formateur :

=====

Félicitations pour votre participation !

=====

Conclusion

- Beaucoup de choses possible
 - Si envie de commencer, pas si compliqué
 - Au niveau de l'ESR, enjeu de se grouper pour demander des choses
 - En cours pour l'infra avec *IaaS*
 - Serait intéressant que les utilisateurs se regroupent aussi pour échanger, pour demander des choses...
 - IA pousse quelque part à devenir un ingé senior
 - avec compétences : spécifications, orchestrateur de tâches, vérification et intégration
- Marche pas facile à passer pour des développeurs débutants et intermédiaires

Merci de votre attention !

Annexe : les benchmarks

Leaderboard Aider : <https://aider.chat/docs/leaderboards/>

LiveBench : A Challenging, Contamination-Free LLM Benchmark
<https://livebench.ai/#/?Coding=a>

SWE Arena: An Open Evaluation Platform for Automated Software Engineering
<https://livebench.ai/#/?Coding=a>

SWE Lancer : <https://arxiv.org/abs/2502.12115>

Leaderboard WebDev Arena : <https://web.lmarena.ai/leaderboard>