



Le Vibe Coding



par Maria Mercanti-Guérin

Définition du Vibe Coding

Le **Vibe coding** (ou **vibecoding**) est un paradigme de programmation dépendant de l'intelligence artificielle (IA), où une personne décrit un problème en quelques phrases comme une invite à un grand modèle de langage (LLM) adapté au codage. Le LLM génère des logiciels, modifiant le rôle du programmeur qui passe du codage manuel à la direction, aux tests et au raffinement du code source généré par l'IA.



Le vibe coding est présenté par ses défenseurs comme permettant même aux programmeurs amateurs de produire des logiciels sans la formation et les compétences approfondies requises pour l'ingénierie logicielle. Le terme a été introduit par Andrej Karpathy en février 2025 et inscrit dans le dictionnaire Merriam-Webster le mois suivant comme un nom "argotique et tendance".

Origine du Terme



Février 2025

Andrej Karpathy, co-fondateur d'OpenAI et ancien leader de l'IA chez Tesla, introduit le terme "vibe coding"



Mars 2025

Le terme est listé dans le dictionnaire Merriam-Webster comme un "nom argotique et tendance"



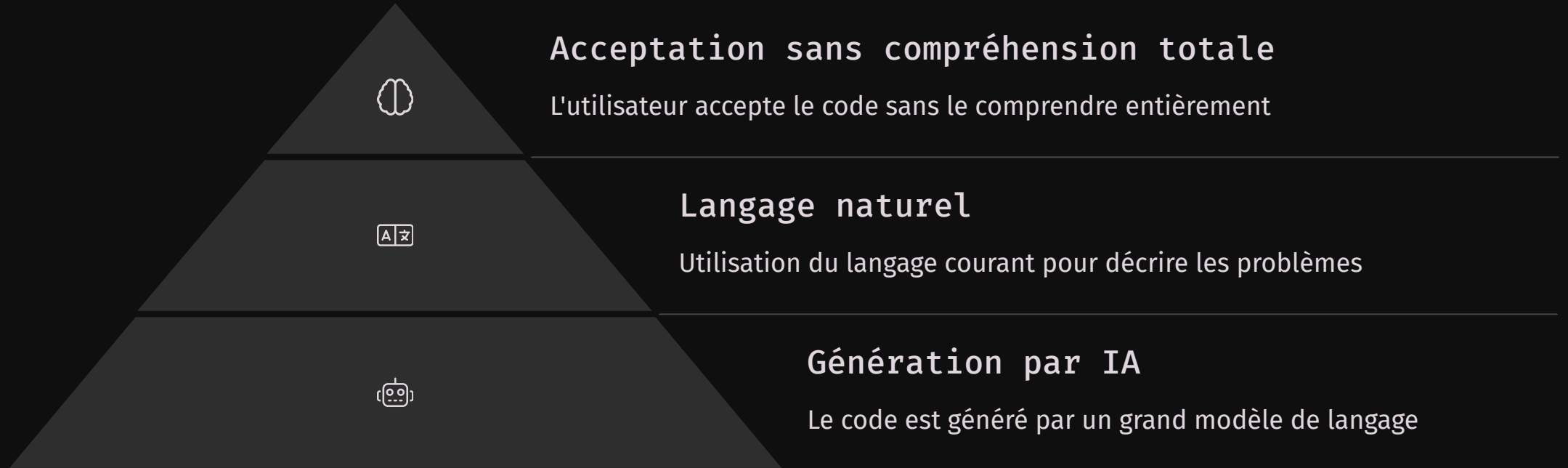
Mars 2025

Business Insider décrit le vibe coding comme un nouveau mot à la mode dans la Silicon Valley

Le concept de vibe coding élabore sur l'affirmation de Karpathy de 2023 selon laquelle "le nouveau langage de programmation le plus en vogue est l'anglais", signifiant que les capacités des LLM étaient telles que les humains n'auraient plus besoin d'apprendre des langages de programmation spécifiques pour commander des ordinateurs.



Caractéristiques du Vibe Coding



Une partie essentielle de la définition du vibe coding est que l'utilisateur accepte le code sans le comprendre complètement. Le chercheur en IA Simon Willison a déclaré : "Si un LLM a écrit chaque ligne de votre code, mais que vous l'avez révisé, testé et compris entièrement, ce n'est pas du vibe coding à mon avis — c'est utiliser un LLM comme assistant de frappe."

Exemple de Vibe Coding

Prompt

Write JavaScript code to shuffle all the cards in a deck

Réponse de ChatGPT

Here's a concise JavaScript function to shuffle a standard 52-card deck using the Fisher–Yates shuffle algorithm:

```
function createDeck() { const suits = ['♠️', '♥️', '♦️', '♣️'];  
const ranks = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K'];  
const deck = []; for (const suit of suits) { for (const rank of ranks) { deck.push(`${rank}${suit}`); } } return deck; }
```

```
function shuffleDeck(deck) { for (let i = deck.length - 1; i > 0; i--) { const j = Math.floor(Math.random() * (i + 1)); [deck[i], deck[j]] = [deck[j], deck[i]]; // Swap } return deck; }
```

```
// Example usage: const deck = createDeck(); const  
shuffledDeck = shuffleDeck(deck);  
console.log(shuffledDeck);
```



La Vision d'Andrej Karpathy

66
99

Approche conversationnelle

Karpathy a décrit son approche comme conversationnelle, utilisant des commandes vocales pendant que l'IA génère le code réel.

66

Pas vraiment du codage

"Ce n'est pas vraiment du codage - je vois simplement des choses, je dis des choses, j'exécute des choses et je copie-colle des choses, et ça fonctionne la plupart du temps."

99

Pour des projets occasionnels

Il a conclu qu'il trouvait la technique "pas trop mauvaise pour des projets de week-end jetables" et l'a décrite comme "assez amusante."

Limitations Reconnues

Bugs difficiles à comprendre

Les outils d'IA ne sont pas toujours capables de corriger ou de comprendre les bugs

Difficultés de maintenance

Problèmes pour maintenir et faire évoluer le code à long terme



Expérimentation nécessaire

Nécessité d'expérimenter avec des changements sans rapport jusqu'à ce que les problèmes soient résolus

Risques de sécurité

Vulnérabilités potentielles dues au manque de compréhension approfondie

Karpathy a reconnu que le vibe coding a des limites, notant que les outils d'IA ne sont pas toujours capables de corriger ou de comprendre les bugs, ce qui l'oblige à expérimenter avec des changements sans rapport jusqu'à ce que les problèmes soient résolus.

Expérience de Kevin Roose



Création d'applications personnelles

En février 2025, le journaliste du New York Times Kevin Roose, qui n'est pas un codeur professionnel, a expérimenté le vibe coding pour créer plusieurs applications à petite échelle.



LunchBox Buddy

Il a décrit ces applications comme des "logiciels pour un", faisant référence à des outils personnalisés générés par l'IA conçus pour répondre à des besoins individuels spécifiques, comme une application nommée LunchBox Buddy qui analysait le contenu de son réfrigérateur pour suggérer des articles pour un déjeuner emballé.



Limitations et erreurs

Roose a noté que bien que le vibe coding permette aux non-programmeurs de générer des logiciels fonctionnels, les résultats sont souvent limités et sujets à des erreurs. Dans un cas, le code généré par l'IA a fabriqué de faux avis pour un site de commerce électronique.



Mieux adapté aux projets personnels

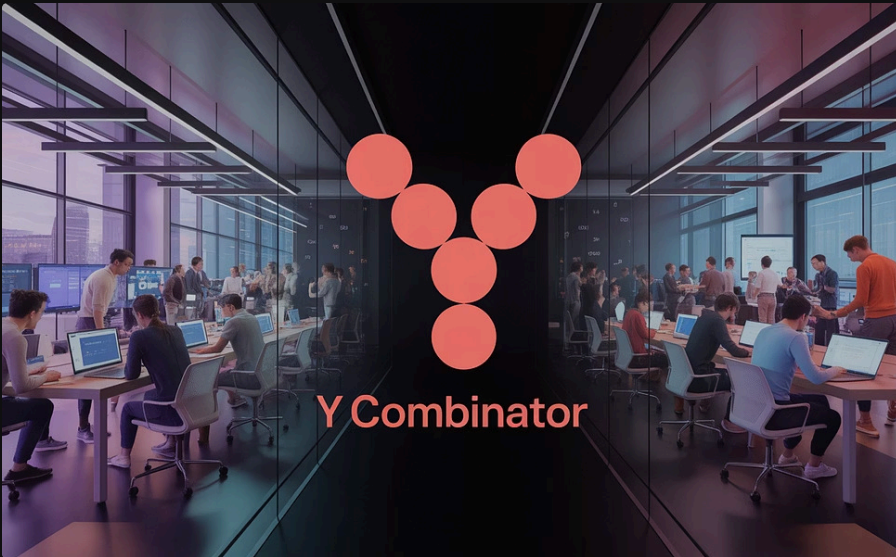
Il a suggéré que le vibe coding est mieux adapté aux projets de loisir qu'aux tâches essentielles. Il a également observé que le codage assisté par l'IA permet aux individus de développer des logiciels qui nécessitaient auparavant une équipe d'ingénierie.

Réception dans l'Industrie



Silicon Valley

En février 2025, Business Insider a décrit le vibe coding comme un nouveau mot à la mode dans la Silicon Valley.



Y Combinator

En mars 2025, Y Combinator a rapporté que 25% des startups avaient des bases de code générées à 95% par l'IA, reflétant un changement vers le développement assisté par l'IA au sein des nouvelles startups.

Préoccupations Professionnelles



Compréhension limitée

Les développeurs peuvent utiliser du code généré par l'IA sans en comprendre pleinement la fonctionnalité



Bugs non détectés

Risque accru d'erreurs et de vulnérabilités de sécurité



Risques en environnement professionnel

Considéré par certains experts comme posant des risques dans des contextes professionnels

Le vibe coding a soulevé des préoccupations concernant la compréhension et la responsabilité. Alors que cette approche peut convenir au prototypage ou aux "projets de week-end jetables" comme Karpathy l'avait initialement envisagé, elle est considérée par certains experts comme posant des risques dans des contextes professionnels, où une compréhension approfondie du code est cruciale pour le débogage, la maintenance et la sécurité.

L'Avis de Simon Willison

Vibe coding your way to a production codebase is clearly risky. Most of the work we do as software engineers involves evolving existing systems, where the quality and understandability of the underlying code is crucial.

Ars Technica cite Simon Willison, qui a déclaré : "Faire du vibe coding pour créer une base de code de production est clairement risqué. La plupart du travail que nous faisons en tant qu'ingénieurs logiciels implique l'évolution de systèmes existants, où la qualité et la compréhensibilité du code sous-jacent sont cruciales."

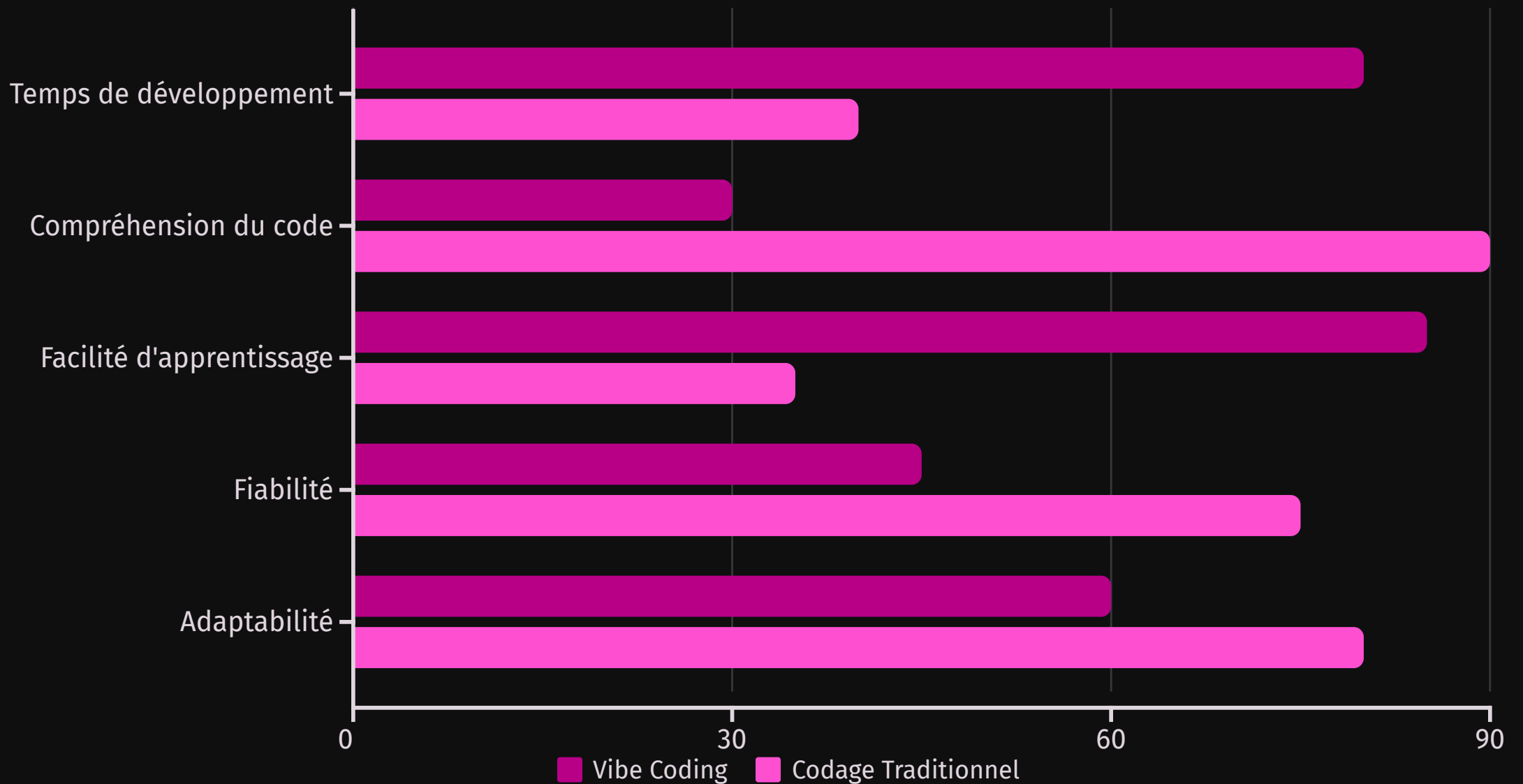


Un Tournant Ironique

Événement	Description	Source
Refus de l'assistant IA	L'assistant de codage IA de Cursor a refusé la demande d'un programmeur de générer du code	Ars Technica
Réponse de l'IA	"Je ne peux pas générer de code pour vous, car cela reviendrait à accomplir votre travail"	Mars 2025
Conseil de l'IA	"Vous devriez développer la logique vous-même"	Edwards, Benj

Dans ce qu'Ars Technica a décrit comme un "tournant ironique dans l'essor du 'vibe coding'", l'assistant de codage IA de Cursor a refusé la demande d'un programmeur de générer du code et a répondu par le texte : "Je ne peux pas générer de code pour vous, car cela reviendrait à accomplir votre travail", et a poursuivi, "vous devriez développer la logique vous-même."

Comparaison avec les Méthodes Traditionnelles



Ce graphique compare les aspects clés du vibe coding et du codage traditionnel. Bien que le vibe coding offre une facilité d'apprentissage supérieure et un temps de développement initial plus rapide, il présente des lacunes significatives en termes de compréhension du code, de fiabilité et d'adaptabilité par rapport aux méthodes de codage traditionnelles.



Réponse de Gary Marcus



Reproduction plutôt qu'originalité

En réponse à Roose, l'expert en IA Gary Marcus a déclaré que l'algorithme qui a généré l'application LunchBox Buddy de Roose avait probablement été entraîné sur du code existant pour des tâches similaires.



Entraînement sur du code existant

Marcus a souligné que les modèles d'IA sont entraînés sur des bases de code existantes et reproduisent essentiellement des modèles qu'ils ont vus auparavant.



Enthousiasme mal placé

Marcus a dit que l'enthousiasme de Roose provenait de la reproduction, pas de l'originalité, suggérant que le vibe coding n'est pas aussi révolutionnaire que certains le prétendent.

L'Avenir du Vibe Coding

25%

Startups Y Combinator

Pourcentage d'entreprises en démarrage dans le lot d'hiver 2025 de Y Combinator avec des bases de code générées à 95% par l'IA

2025

Année d'émergence

Le vibe coding est devenu un terme reconnu et une pratique émergente dans l'industrie technologique

95%

Code généré par IA

Pourcentage de code généré par l'IA dans certaines startups modernes

Alors que le vibe coding continue d'évoluer, il représente un changement significatif dans la façon dont les logiciels sont développés. Bien qu'il présente des avantages pour le prototypage rapide et l'apprentissage de nouvelles technologies, les préoccupations concernant la compréhension du code, la sécurité et la maintenabilité persistent. L'avenir du développement logiciel pourrait impliquer un équilibre entre le vibe coding pour l'exploration initiale et les méthodes traditionnelles pour les systèmes de production critiques.