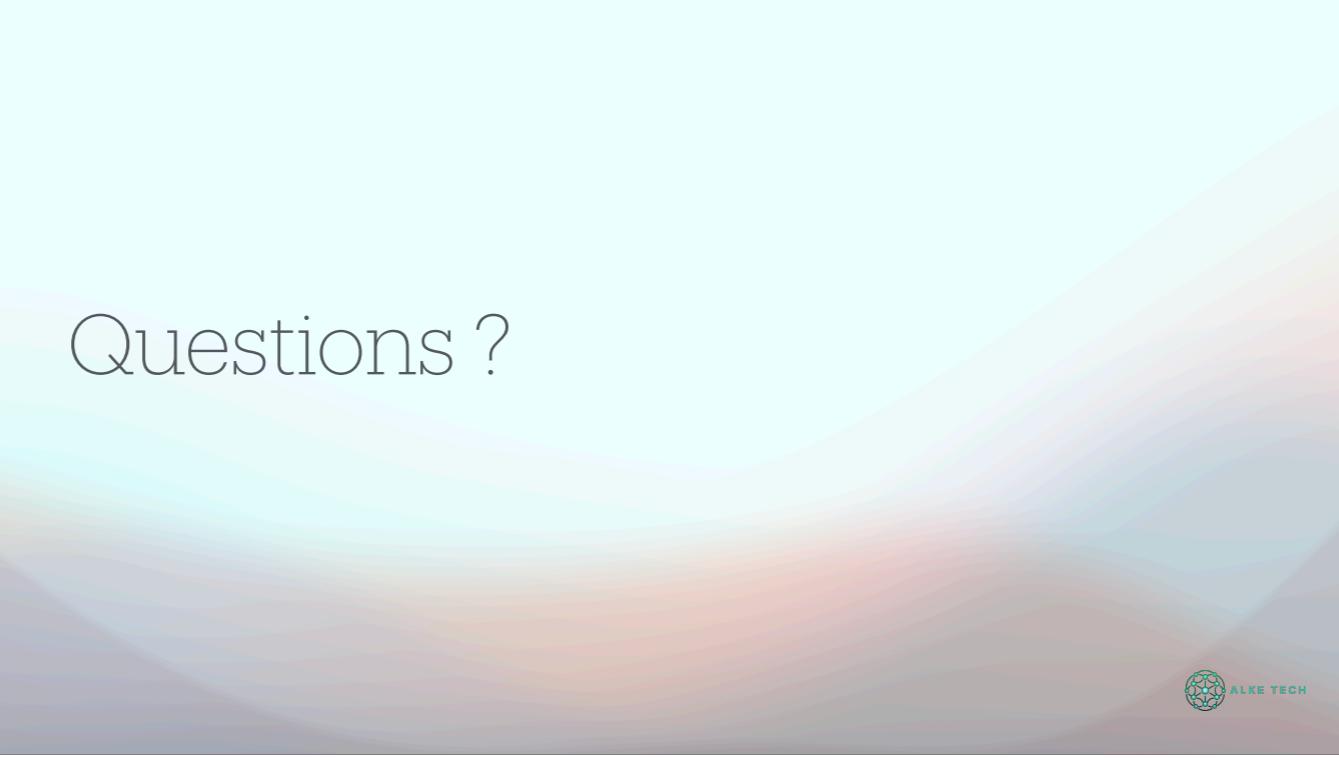


Comment être un(e) bon(ne) dév ?

À l'heure des IA génératives

Xavier Leune





Questions ?



Est-ce que vous avez des questions ? Ah non pardon

Questions !



C'est moi qui ai des questions pour vous ! Pour commencer je vous propose qu'on inverse un peu le déroulé traditionnel, je vais vous poser quelques questions auxquelles vous pouvez répondre à main levée.

Qui a connu PHP 8.0 ? PHP 7.0 ? L'époque où PHP 6 devait sortir ? Et finalement PHP 5.3 ? Qui a connu Silverlight ? Adobe Air ? Adobe Flash ?

Qui a connu Dreamweaver ? Microsoft FrontPage ?

Qui a déployé avec des FTP ? Qui a codé directement en prod avec des FTP ?

Qui a connu l'époque où l'usage de javascript et de css avec du html faisait qu'on appelait ça du DHTML ?

J'ai commencé à faire du dév il y a 22 ans et ça fait 18 ans que c'est mon métier, j'ai vu passer énormément de technos - j'en ai cité quelques unes avec lesquelles je me suis amusé, pratiqué plus ou moins longtemps. La plupart de ces technos sont aujourd'hui mortes ou alors leurs usages ont complètement évolué et pourtant c'était énormément déployé. Tout ça me permet de me rappeler que :

La seule constante dans notre métier, c'est le changement.

Pour les prochaines questions je vais passer parmi vous et si vous êtes ok je proposerai à quelques personnes de nous donner également leur point de vue.

- Qui a essayé des LLM pour écrire du code ?
- Qui utilise l'IA au quotidien pour écrire du code ? Par exemple avec Claude code ou Jetbrains Junie ou autre ?
 - Tu peux nous donner ton prénom et depuis combien d'années tu fais ce métier ?
 - Qu'est-ce que tu en pense ?
 - Est-ce que tu penses que ça t'aide à être meilleur dans ton métier ? Pourquoi ?
- Qui a essayé mais a finalement arrêté ?
 - Tu peux nous donner ton prénom et depuis combien d'années tu fais ce métier ?

- Pourquoi tu as arrêté ?
- Quelqu'un d'autre souhaite intervenir pour donner son avis ?

Xavier Leune
Fondateur



ALKE TECH

Empowering publishers with technology

www.alketech.eu
linkedin.com/in/xleune
@beoneself.bsky.social

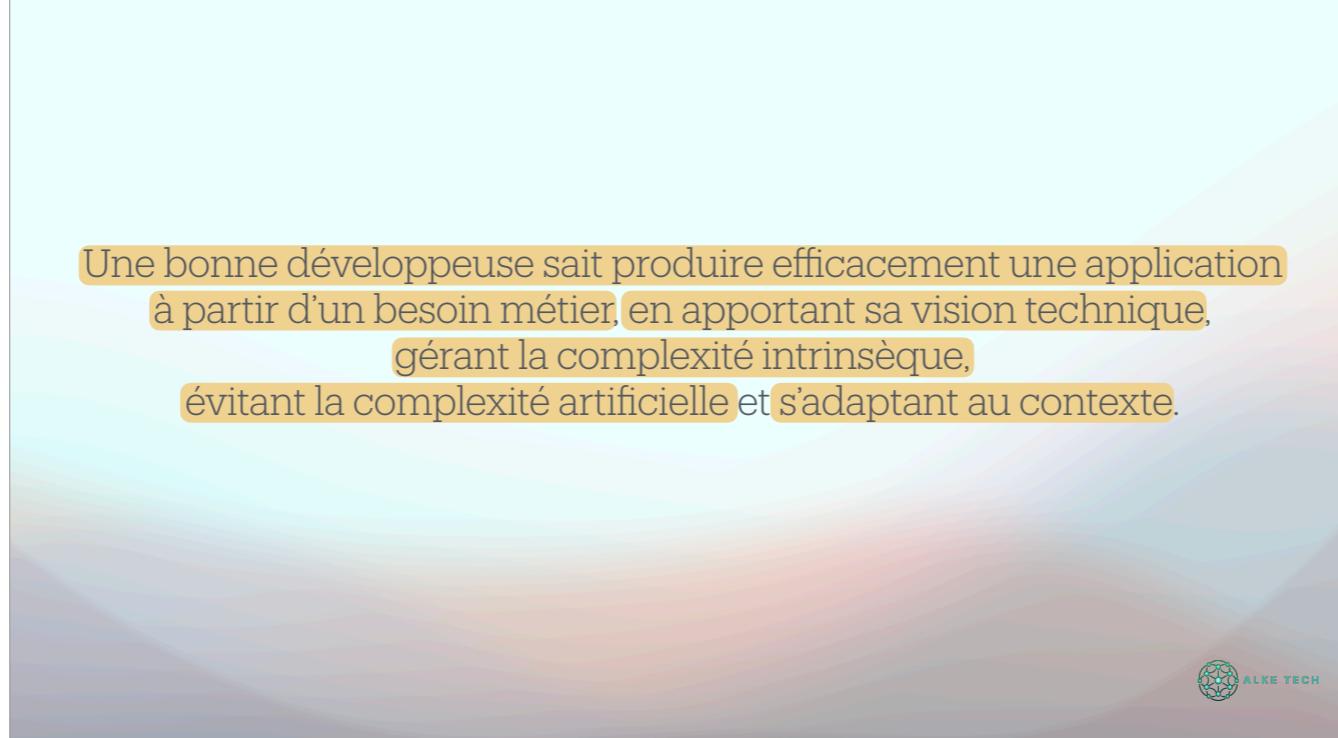


Je m'appelle Xavier Leune, j'ai 18 ans d'expérience en PHP et je suis fondateur d'Alke Tech, où je construit le futur des outils d'analytics.

Au cours de ma carrière j'ai été responsable de dizaines de recrutements faisant passer des 100aines d'entretiens, devant décider si telle ou telle personne se révélerait être un bon ou une bonne dév une fois intégré, d'évaluations annuelles, devant déterminer qui avait le mieux progressé au cours de cette année et in fine déterminer les augmentations, décider parfois qui ne convenait pas dans ce contexte.

J'ai donc été obligé de me forger ma propre opinion sur ce qui fait un bon développeur ou une bonne développeuse.

Et en préparant cette conf j'ai d'abord commencé par chercher une définition qui ferait consensus, mais finalement je n'ai rien trouvé de concis. Je vais donc vous proposer ma propre définition.



Une bonne développeuse sait produire efficacement une application
à partir d'un besoin métier, en apportant sa vision technique,
gérant la complexité intrinsèque,
évitant la complexité artificielle et s'adaptant au contexte.



Découpons cette définition:

- **Une bonne dév sait produire efficacement une application:** Le mot clé ici, c'est pas application (qui englobe toutes les formes d'apps, de la CLI à la lib, à l'app web, ...): c'est efficacement. dépenser le moins de ressource possible pour le meilleur résultat.
- **à partir d'un besoin métier:** c'est assez limpide: on travaille pour le business avant tout, pour avoir de l'impact. Le code pour le code ne vaut rien.
- **En apportant sa vision technique:** Pour y arriver correctement et maximiser notre impact, les échanges avec le business sont essentiels: ça permet d'ajuster la manière de répondre au besoin, trouver les meilleurs compromis techniques et fonctionnels. Il y a un besoin de communication important sur cette partie.
- **Gérant la complexité intrinsèque:** c'est celle qui vient du business. Les règles métiers peuvent être compliquées, on ne peut s'y dérober, c'est à nous de trouver les bonnes solutions pour que ça ne devienne pas un cauchemar. **Si vous gérez le site des impôts, vous devez gérer tous les modes de calculs de l'impôt, avec toutes les combinaisons possibles. Pas une partie.**
- **Évitant la complexité artificielle:** La complexité artificielle c'est celle qu'on produit nous même. Qui ne répond à aucun besoin business. Par exemple: utiliser une architecture micro services lorsqu'un monolithe suffit.
- **S'adaptant au contexte:** Là encore le pragmatisme est clé. Il y a une équipe ? Alors il faut savoir communiquer avec elle. La boite démarre tout juste ? Alors peut-être que la clé du succès sera dans votre capacité à livrer vite. Votre projet est bien établi et chaque bug coûte cher ? Alors sans doute que les tests et la qualité primeront sur votre délai de livraison. « There is no silver bullet » Il n'y a pas de réponse magique dans le dév, tout ne convient pas à toutes les organisations. On peut être très bon dans un contexte et beaucoup moins bon dans un autre.

Acte I: Les faits



Le passé



Je ne sais pas vous, mais j'ai l'impression que le coup de la techno qui va remplacer les dévs, on nous l'a déjà faite.

Visual Basic

va permettre de créer des applications sans coder.

Années 80



Delphi

va permettre de créer des applications sans coder.

Années 80



PowerBuilder

va permettre de créer des applications sans coder.

Années 80



Dreamweaver

va permettre de créer des applications sans coder.

Années 90/2000



UML

va permettre de créer des applications sans coder.

Années 90/2000



WordPress

va permettre de créer des applications sans coder.

Années 2010



*Outil No-Code

va permettre de créer des applications sans coder.

Années 2010



Zapier / Make / N8N

va permettre de créer des applications sans coder.

Années 2010

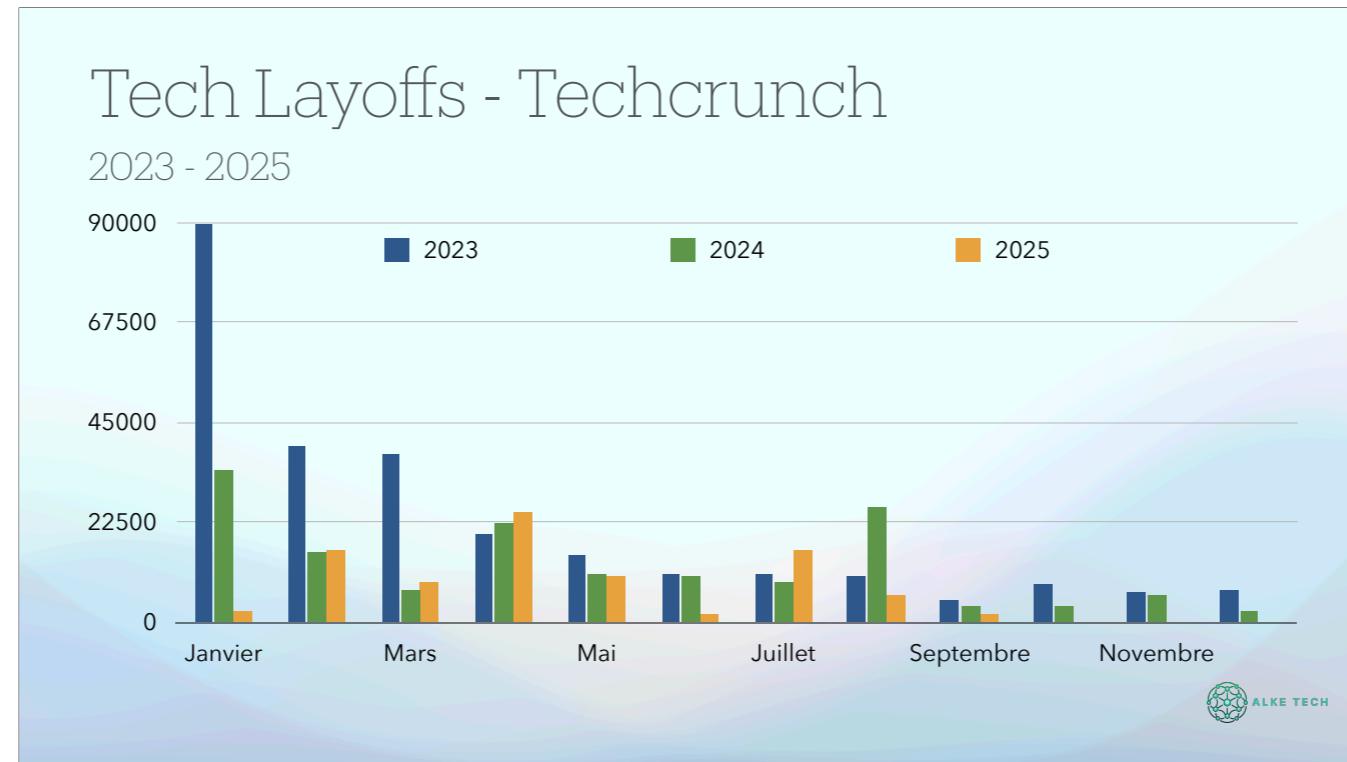


Github Copilot / ChatGPT / Claude Code / ...

va permettre de créer des applications sans coder.

Années 2022+





Blague à part, ces années ont tout de même été compliquées. Il y a eu beaucoup de licenciements dans la tech depuis 2023 et la tendance ne semble pas au ralentissement. D'énormes investissements sont réalisés par les plus grosses sociétés dans l'IA et elles ont besoin de libérer du cash flow en licenciant de grandes quantités de personnes. Quand on parle de licenciements dans la tech, on parle de tout l'écosystème, pas uniquement de dévs... Cependant...

<https://techcrunch.com/2025/09/22/tech-layoffs-2025-list/>

<https://techcrunch.com/2024/12/31/a-comprehensive-archive-of-2024-tech-layoffs/>

<https://techcrunch.com/2024/05/01/a-comprehensive-archive-of-2023-tech-layoffs/>

40%

Part des dévs dans les licenciements chez Microsoft - 2025

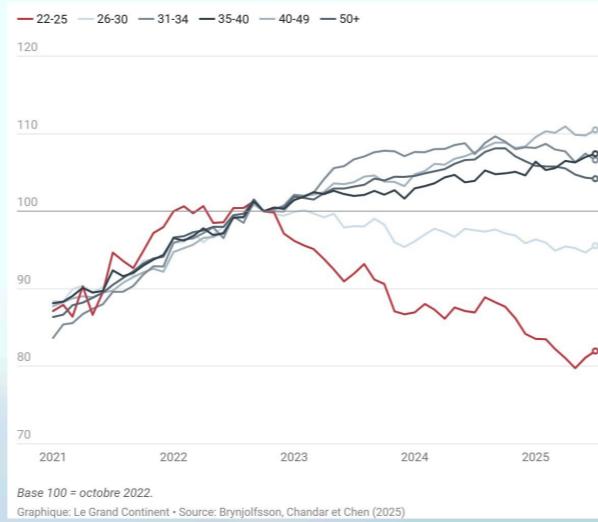


<https://techcrunch.com/2025/05/15/programmers-bore-the-brunt-of-microsofts-layoffs-in-its-home-state-as-ai-writes-up-to-30-of-its-code/>

<https://www.finalroundai.com/blog/ai-tech-layoffs-mid-2025>

Effectif des dév logiciel

USA - Par tranche d'âge



Attention avec ce graphe qui circule partout en ce moment:

- Commence après la forte croissance covid
- Indexé sur l'âge et non sur l'expérience
- Peut-être un effet de « retour à la normale » ou de la fin de l'attractivité des formations bootcamp
- Lien potentiel avec l'IA, à prendre avec des pincettes

Pour rappel date de sortie de chatgpt c'est novembre 22, c'est la base 100 du graphique. À ce moment là pour le code c'était encore vraiment pas ouf. Donc mega warning. Si je devais conclure je dirais que seule la 2nde baise, entre 24 et 25 est à attribuer à l'IA.

Nombre de dévs dans le monde

Étude Jetbrains - en millions



https://www.jetbrains.com/fr-fr/lp/devcosystem-data-playground/#global_population

On attend les chiffres de 2025 mais c'est intéressant de mettre en perspective les 2 évolutions en sortant du centrage USA.



Le présent



L'impact sur la productivité

Ce que les études disent

Étude	Méthode	Gain de productivité
Becker & Rush et al. (2025) - METR	16 dévs - Open source	- 19 %
Peng et al. (2023) - GitHub Copilot	95 dévs - Freelances	+ 55,8 %
Paradis et al. (2024) - Google	96 dévs - Interne	+ 56 %
Cui et al. (2025) - Microsoft, Accenture, Fortune 100	4 867 dévs - 3 entreprises	+ 26%



On va mettre tout de suite les pieds dans le plat. Toutes les grosses études montrent une hausse importante de la productivité.

Toutes les études sont citées à partir de celle-ci: <https://arxiv.org/abs/2410.12944>

Même si la mesure de la productivité c'est compliqué, qu'on a tendance à se la jouer un peu artistes à ce niveau là et à refuser ce genre de mesure, cependant c'est important même si c'est imparfait.

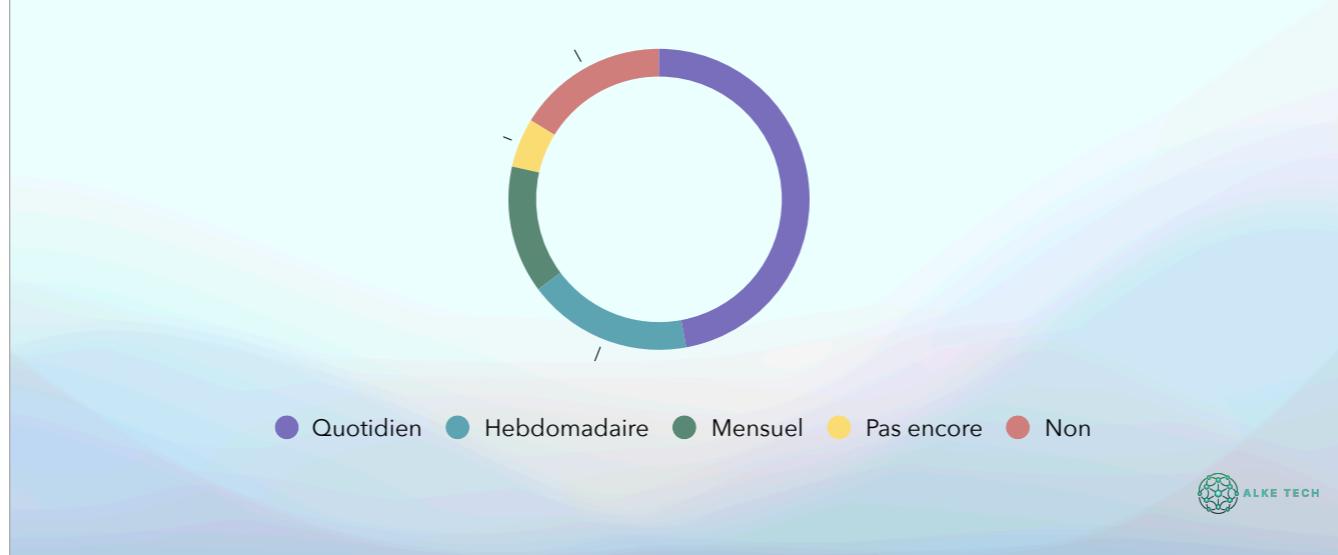
49 000

Réponses au StackOverflow Survey 2025



ALKE TECH

Usage des LLM par les dévs



<https://survey.stackoverflow.co/2025/ai#1-ai-tools-in-the-development-process>

+ 49K Dév ont répondu à l'édition 2025 du sondage de Stack overflow. Large part consacrée aux LLM

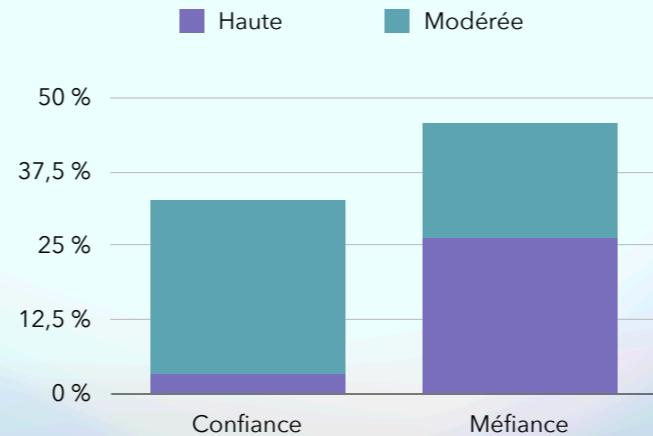
+ Voici la part des dévs qui utilisent

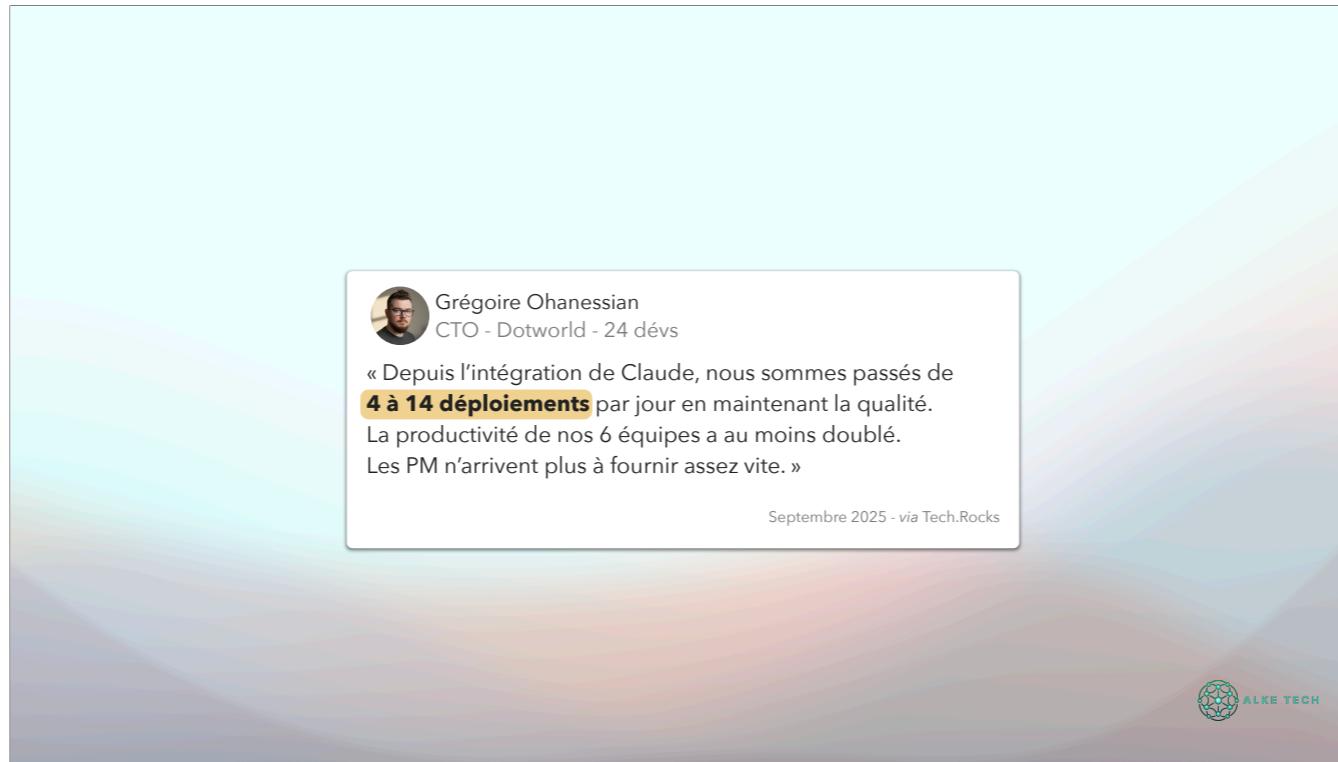
Usage des LLM par les dévs



+ Ça fait 79% d'utilisateurs de l'IA parmi les dévs.
C'est énorme comme adoption. Cependant cette

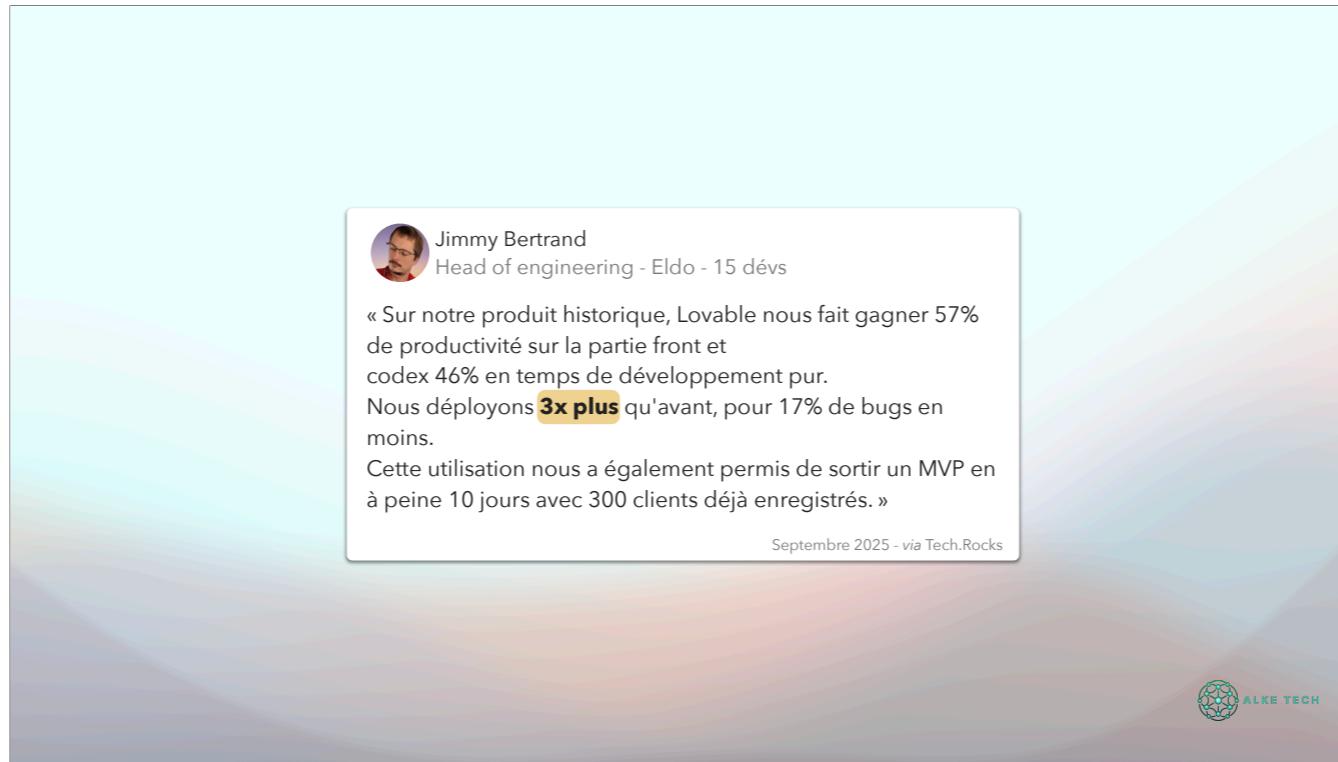
La confiance dans les LLM





Gain de productivité au moins x 2. C'est à prendre avec des pincettes car **c'est difficile à évaluer sérieusement.**

Gros avantage sur les revues de code effectuées par l'ia et effet positif: les dévs font beaucoup + gaffe en amont. **Faut croire que c'est pas agréable de se faire reprendre en public par un robot, même s'il est poli.**



Lovable pour les PM. Gros gain de temps vs figma. Maquette fonctionnelle pour dès qu'il y a plus qu'un composant.

57% de temps de dév gagné grâce à lovable.

Que des seniors.

La boite a été créée en 2016. Grosse levée 2020 / 2021.

Nombre de mep depuis codex: x3.

Rédaction commerciale / story: 46% de productivité.

80/90% de review codex étaient utiles à prendre en compte avant MEP.

En quelques mois le nombre de bug a vachement diminué. -17%, c'était un gros sujet.

Culture du micro saas depuis peu (pros du batiment): simulation visuelle avec nano banana lors de la sortie ; en 8 jours c'était bon. 300 clients dessus, complètement vibe codé. Puis migration vers équipe de dév pour reprendre ça propre.

2ème micro saas en cours.

Le PM peut avoir du mal à suivre. Grosse pression pour lui.

 Pierre Maréchal
CTO - Digitaleo - 30 dévs

« Nous avons commencé le déploiement au début de l'été. Les résultats sont bons sur les monorepo, mais il va nous falloir adapter nos micro services pour que l'agent s'y retrouve.

Nous avons 2 points d'attention:

La **complexité du code** peut augmenter face à la gestion de cas pas toujours utiles et

La **charge mentale** lorsqu'on travaille avec plusieurs agents simultanément. »

Septembre 2025 - via Tech.Rocks



A démarré avec cursor / copilot. Un peu en retard ces derniers temps, déploiement massif de claude code mais test de codex & windsurf.

Très récent: démarrage courant de l'été. Structuration depuis septembre.

Ça prend assez lentement. Archi microservices avec plein de repos, ça n'est pas évident pour utiliser claude code.

REX toutes les semaines au sein de l'équipe.

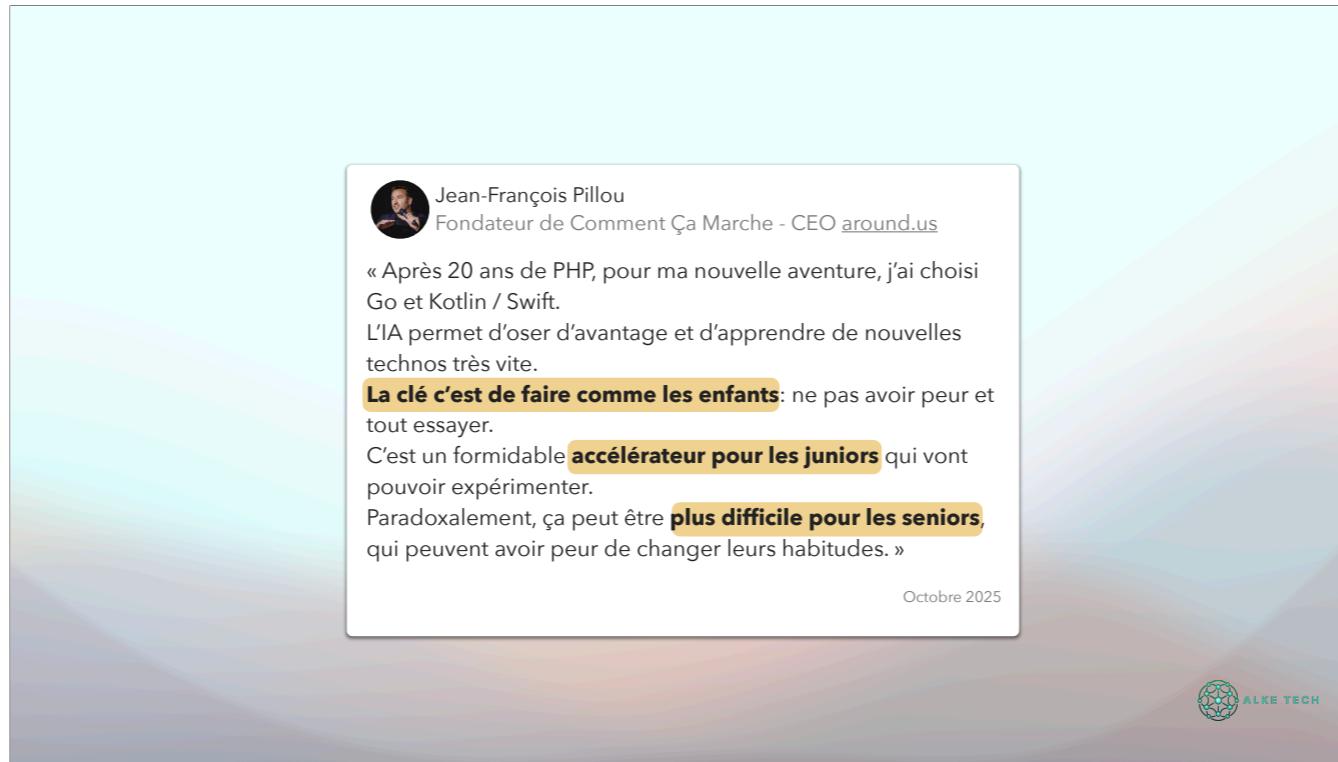
Plusieurs approches: + de tests, review auto, backo interne (monorepo): ça marche très bien & permet d'aller très vite.

Pain point: tu lances +rs agents, tu peux t'ennuyer pendant que ça bosse. Mais charge mentale beaucoup + importante et ça épouse les équipes. Attention à ce point.

Attention aussi parfois à la complexité supplémentaire que ces outils apportent: en gérant + de cas (parce qu'il y pense et que t'y aurais pas pensé).

Un peu tôt pour avoir des feedbacks sur la productivité.

Intéressant:



Invité à la WWDC et around.us passé dans la sélection de l'apple store.

Connexion directe business <-> dev: 1 seule personne

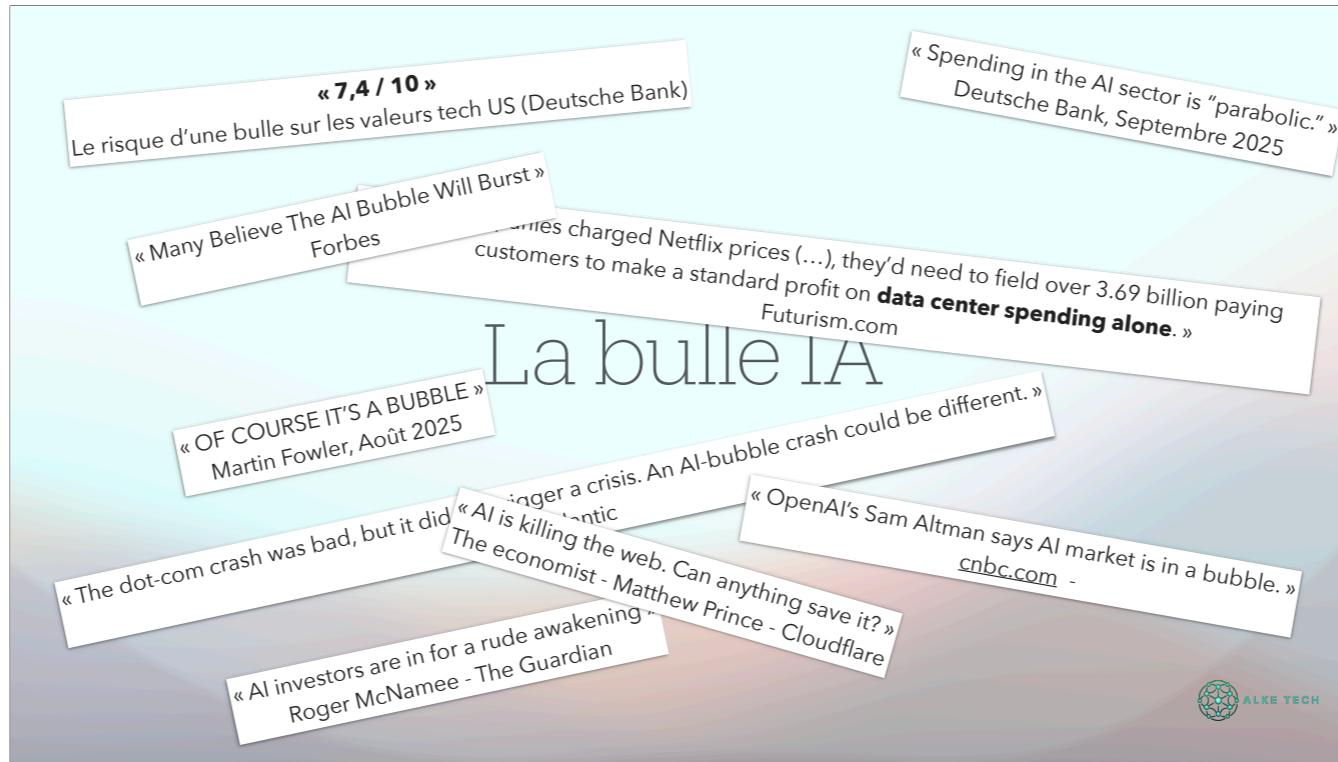
Probablement de + en + de cas similaires à l'avenir.

La meilleure approche pour tirer partie des LLM: bien respecter a minima SOLID. Le LLM s'y retrouvera mieux et le dév aussi.



Le futur





<https://fortune.com/2025/09/23/ai-boom-unsustainable-tech-spending-parabolic-deutsche-bank/>

<https://martinfowler.com/articles/202508-ai-thoughts.html>

<https://futurism.com/data-centers-financial-bubble>

<https://www.theatlantic.com/economy/archive/2025/09/ai-bubble-us-economy/684128/>

<https://www.theguardian.com/commentisfree/2025/sep/24/ai-investors-llms>

<https://www.cnbc.com/2025/08/18/openai-sam-altman-warns-ai-market-is-in-a-bubble.html>

<https://www.forbes.com/sites/llewellynking/2025/09/29/many-believe-the-ai-bubble-will-burst-affecting-electric-utilities/>

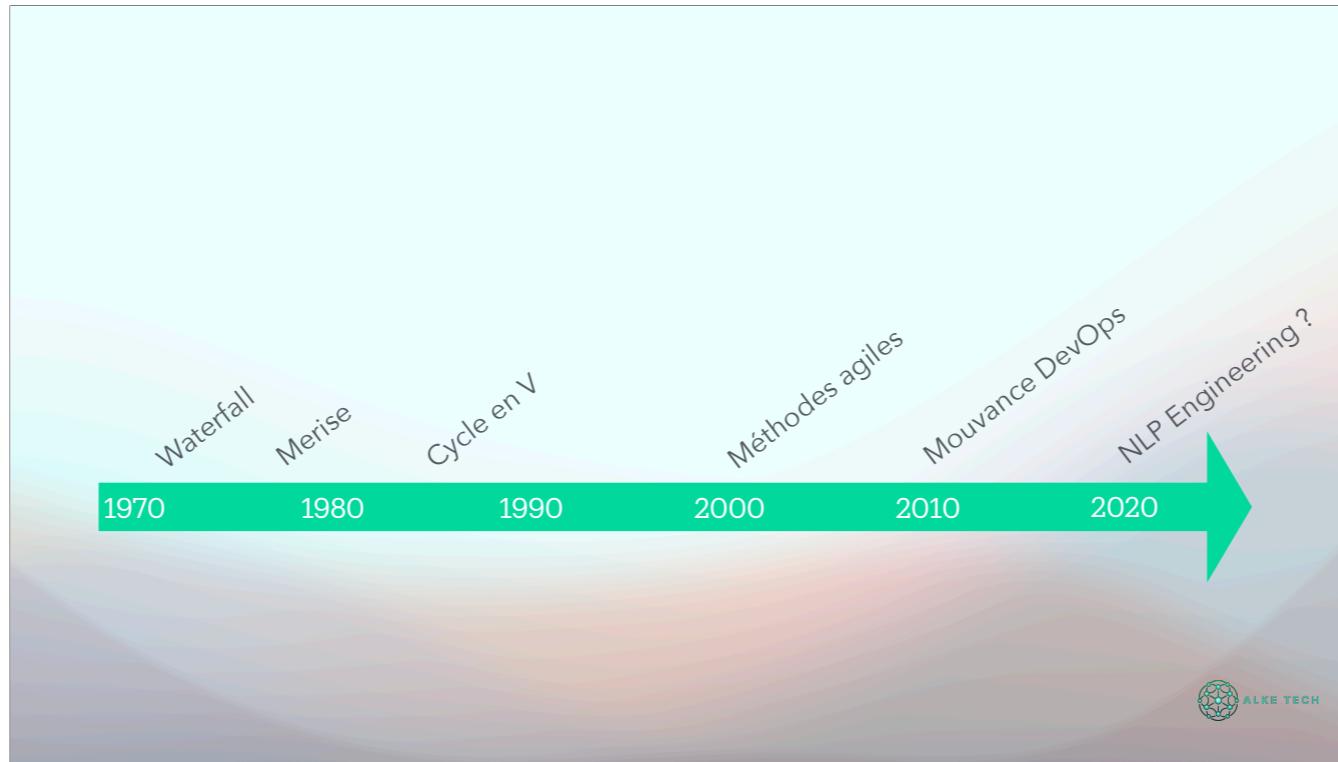
<https://www.techcentral.ie/altman-warns-of-a-repeat-of-the-dot-com-bubble/>

<https://www.economist.com/business/2025/07/14/ai-is-killing-the-web-cananything-save-it>



Acte II: Et nous alors ?





Si on fait l'historique des méthodes d'organisation et de gestion de projet depuis les années 70, on peut constater globalement que progressivement **l'enjeu** est devenu de **raccourcir les cycles de développement et de déploiement**. On peut relier cela au fait que le **coût du développement baisse en continu**, avec l'apparition progressive de **langage de plus haut niveau** qui permettent de gagner du temps, la maturité des communautés autour de ceux-ci avec les **frameworks**. Cela s'applique même à **l'infrastructure** avec l'arrivée de **Kubernetes** qui est avant tout un framework permettant d'industrialiser facilement une infra.

Plus le coût du développement baisse, plus on cherche à raccourcir en miroir les étapes amont. C'est **pertinent de passer 2 mois** à réaliser des spécs et cahier des charges si le moindre module va mettre **6 mois à être développé**. C'est **contre productif si le temps de dév est de 2 semaines**.

L'arrivée de l'IA dans les cycles de développement vient clairement mettre encore plus de pression. Le **code est aujourd'hui presque gratuit**, il n'y a plus de barrière à l'entrée.

Récemment dans Pragmatic Engineer, on apprend **comment Claude Code est développé**. Le cycle est raccourci au maximum avec des itérations sur les prototypes directement à l'aide de Claude Code ; ces prototypes sont tous écrits directement par Claude Code ; testant différentes options en continue jusqu'à ce qu'une soit retenue. Et seulement à ce moment là va être réellement étudié / relu / ajusté.

Cela pourrait devenir la norme, sauf si le prix des token devait massivement augmenter lorsque la bulle IA éclatera.

<https://newsletter.pragmaticengineer.com/p/how-claude-code-is-built>

09 & 10 Octobre 2025
Hotel New-York - TAOM
Disneyland Paris



► event.afup.org



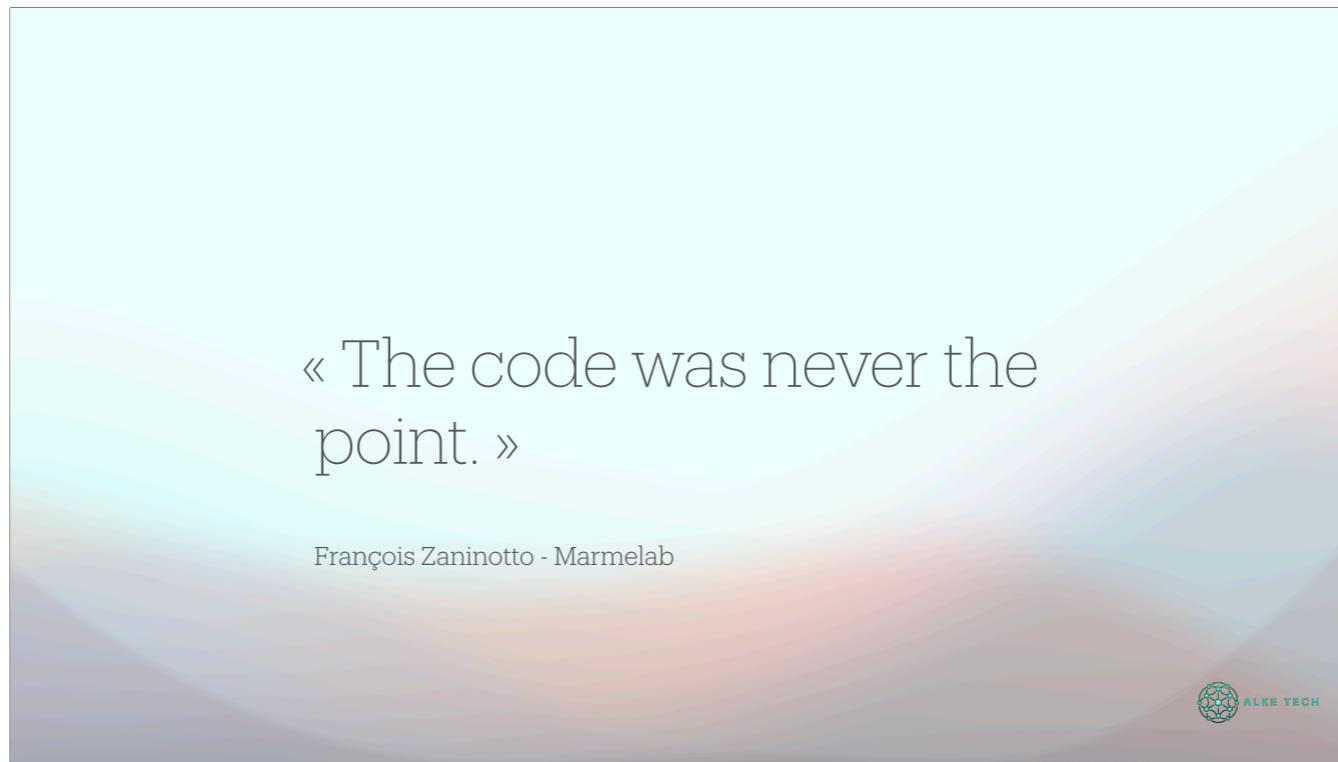
L'ÉVALUATION DES IAS : LA RECETTE SECRÈTE DES AGENTS PAS TROP BÊTES

François ZANINOTTO

Salle Adrien GALLOU - Jeudi - 12h40



Adrien GALLOU - ABCDEF 09/10/2025 12:40-13:00



<https://marmelab.com/blog/2025/09/03/natural-language-developers.html>

Dans cet article très intéressant, François nous livre sa vision de l'impact de l'arrivée des IA sur notre métier.

Il voit notamment 3 types de dévs à l'avenir:

- **Natural Language Devs:** 70% - il produit uniquement à partir de prompts
- **AI Augmented Devs:** 25% - C'est à dire un dév tel qu'il existait pré-chatgpt mais qui embrasse ce changement
- **Foundation Devs:** 5% - Ceux qui dévs les outils d'IA, les drivers, v8 - Qui travaille au plus bas niveau donc

Cette phrase « **The code was never the point** » ouvre la conclusion de cet article. Et **ça résume parfaitement** ce que je pense que l'impact de l'IA sur le dév.

Il n'est pas question de savoir comment est écrit le code - avant les LLM on utilisait déjà de plus en plus de générateurs de code ; l'IA c'est ça mais sans avoir à écrire le générateur.



Don't focus on **output**,
Focus on **outcomes.**



Je vous met ça en anglais car je trouve que plus parlant.

Focus on outcomes, not output. C'est un conseil que j'ai donné plein de fois aux managers que j'encadrais, je ne l'ai pas inventé c'est juste la traduction de ce qu'on attend de vous en entreprise lorsque vous prenez des responsabilités.

L'outcome c'est le résultat de vos actions, l'output c'est leur produit.

La différence entre les 2 est essentielle:

- Output: j'ai redéveloppé le processus d'inscription en page unique
- Outcome: le nombre d'inscription a grimpé de 30% depuis.

Si vous n'avez pas accès à ces informations, demandez les ! Bien souvent les organisations sont ok pour les ouvrir, mais ne le font pas car pensent que ça n'intéressera pas les teams.



N'attendez pas que votre employeur vous forme. Si vous travaillez actuellement dans une entreprise qui interdit le recours à ces outils, il y en a qui le font aujourd'hui pour de bonnes raisons, travaillez avec par vous même. N'oubliez jamais que votre employeur vous formera sur ce qui correspond à ses besoins et ses attentes. Si l'usage des outils d'IA ne rentre pas dedans, vous êtes seul. Sinon dans quelques années ça va être très compliqué pour vous.
Je n'insisterai jamais assez sur ce point: **face à votre carrière et son évolution, vous ne devez compter que sur vous-même.** Vous ne pouvez pas vous reposer sur le volontarisme de quelqu'un d'autre. Vous êtes là aujourd'hui ?



Soyez **responsable**



Vous pouvez utiliser le process que vous voulez, mettre toutes les reviews automatisées, à la fin c'est **vous qui devez garder le contrôle**.

Vous êtes responsable de ce qui passe en prod.

Le code produit par l'IA ne vous plait pas ? Jetez le, recommencez, amenez le, peu importe. Vous devez garder le contrôle de votre application.

« **Knowledge** is knowing that a tomato is a fruit;
Wisdom is not putting it in a fruit salad. »

Miles Kington - Journaliste anglais



La connaissance c'est de savoir que la tomate est un fruit.
La sagesse c'est de ne pas le mettre dans une salade de fruits.

Si on reprend cette analogie, on peut considérer que la **connaissance est apportée par l'outil** - attention raccourci. **A vous d'avoir la sagesse.** C'est à vous de décider ce qui part en prod ou pas, ce qui est une bonne approche ou pas, ce que vous serez capable de maintenir ou pas.

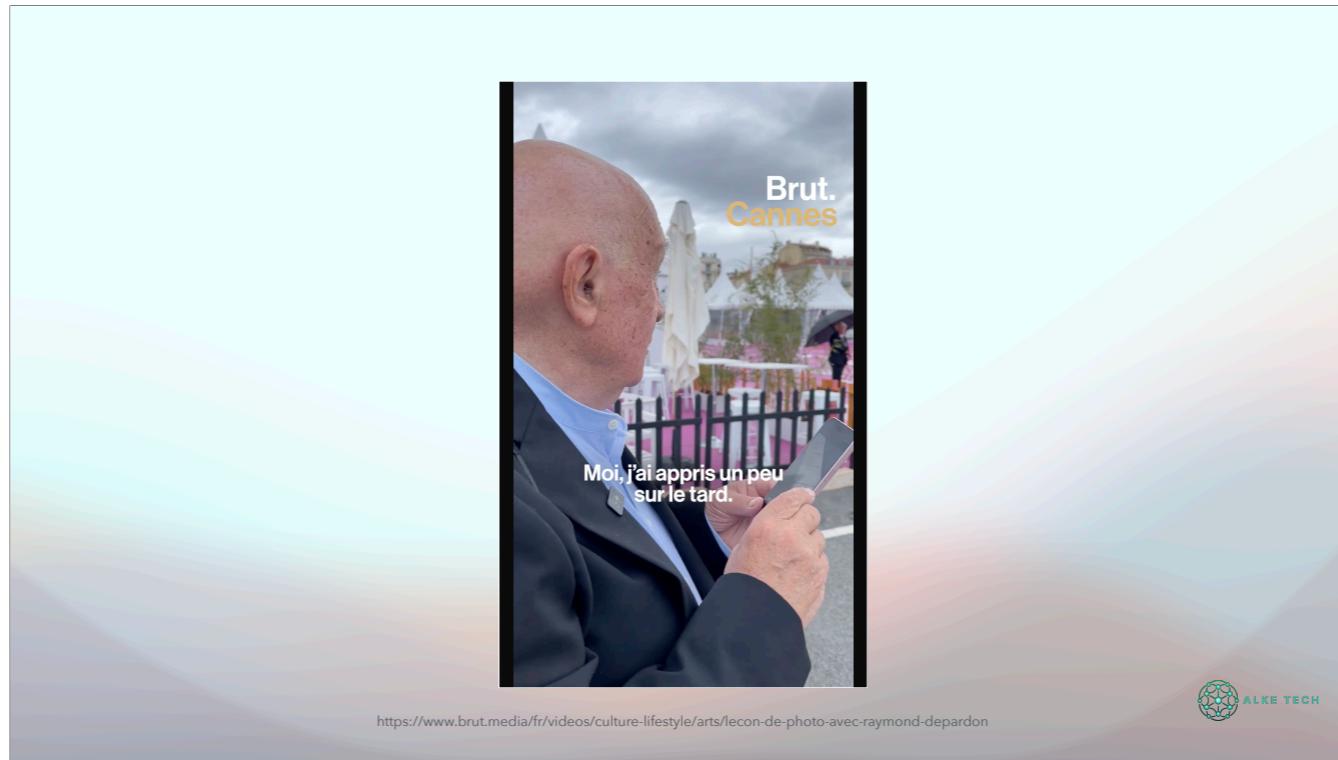


Accueillez les **nouveaux dévs**



Attention au **gatekeeping**. On voit énormément apparaître de commentaires assez dénigrant sur le développement à base d'IA. Cela vient probablement du fait que de nombreuses personnes se sentent menacées par l'arrivée de ces nouvelles pratiques. Je trouve ça personnellement dommage car rien ne nous y oblige.

On va voir arriver dans ce métier plein de personnes qui auront découvert le code à travers l'usage des LLM.



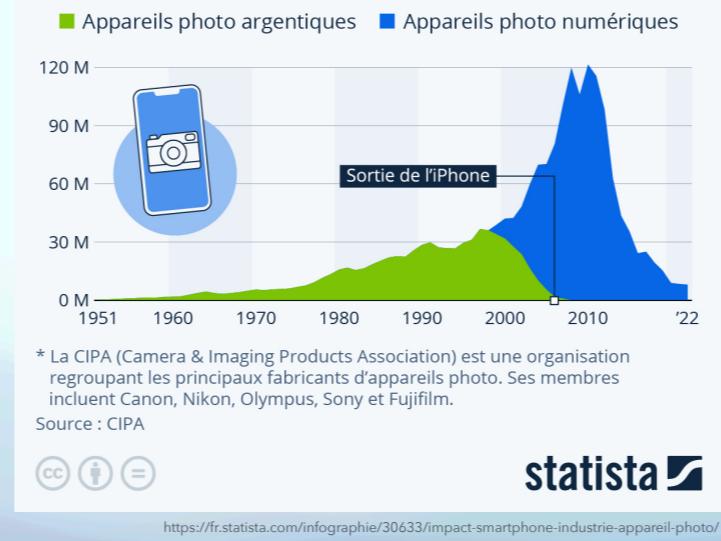
Et c'est possible de faire autrement.

J'ai en tête l'exemple de Raymond Depardon dans cette vidéo réalisée pour Brut. C'est un photographe et réalisateur renommé, qui vient expliquer très simplement, comment chacun peut faire de belles photos avec son smartphone. Et je trouve ça formidable que quelqu'un de son expérience donne ce genre de conseils.

Ou comme dit Orelsan: si t'as envie d'faire un film t'as juste besoin d'un truc qui filme.

Vente d'appareils photo

Impact de la sortie de l'iPhone



L'iPhone est sorti en 2007. Il a démarré la mouvance des smartphones et a littéralement crashé le marché des appareils photo au passage. Par contre la production d'images évoluée de manière exponentielle depuis.

Et au passage l'évolution a été difficile pour les photographes, mais ce métier n'a pas disparu.

Si vous êtes allé à un mariage récemment, vous aurez sûrement vu un photographe pro, peut-être un vidéaste. Si vous avez des enfants scolarisés il y a peu de chance que ça soit un prof qui fasse la photo de classe.

Et s'il est possible de réaliser des films ou séries de qualité avec un smartphone haut de gamme ; ça n'est pas à la portée de la première personne venue. Il faut bien plus que « filmer une scène » pour réaliser un bon film. Cette analogie est proposée par Simon Willison un des créateurs de Django.

Si l'iPhone a démocratisé la photo, les LLM démocratisent le code.



Le code devient **gratuit**,
Il n'a donc plus de **valeur**.



Le code est une étape nécessaire mais pas exclusive pour réaliser une application de qualité.

Le code en lui même n'est pas important. Par contre le contexte dans lequel il est écrit en a énormément. Avec l'arrivé des LLM, on va pouvoir se concentrer sur ce qu'il y a autour du code.

Et le fait que le code n'ait plus de valeur, ça veut dire que si votre seule valeur ajoutée en tant que dév, c'est de produire du code, ça va être très difficile pour vous.



Intéressez vous au **business**



Si vous êtes le genre de dév qui code très bien, mais qui refuse de participer aux réunions de l'équipe, ne s'intéresse qu'au code et pas du tout au business et à l'impact que vous produisez pour votre entreprise, ça va être très compliqué pour vous. Il faut que vous trouviez des axes car notre métier n'est pas de produire du code, c'est de répondre à des besoins. Si selon vous ces besoins sont secondaires, il va falloir que vous soyez 100 x meilleur que l'IA pour pouvoir rentrer en compétition avec elle. Et c'est un jeu que vous ne gagnerez probablement pas.

Les frontières avec le business sont de plus en plus fines, on voit même apparaître des organisations où les dévs sont leurs propres PM, allant encore plus loin dans la vision que **le code n'est qu'un moyen**. Ça ne sera probablement pas le cas partout, cependant si vous ne vous intéressez pas à l'impact de ce que vous produisez, vous ne pourrez plus progresser.



Ne vous **limitez pas** aux LLM



Pour rappel on ne sait pas exactement comment les LLM fonctionnent à l'intérieur. Ça fonctionne, c'est tout. Il y a eu une étude d'Apple très intéressante sur le fait que les LLM n'étaient pas en mesure de créer de nouvelles connaissances, au mieux en synthétiser.

Il y a eu beaucoup de débats d'experts sur la méthodologie et ses conclusions, cependant il y a quelque chose qu'on observe: les LLM ne peuvent pas résoudre tous les problèmes pour vous. Il va falloir encore se retrousser les manches pour un certain nombre de problèmes.

Et c'est tant mieux, car c'est là que **notre expérience** et notre **savoir faire a de la valeur**. Utilisez les à bon escient, mais n'oubliez pas qu'on a besoin de vous pour tout le reste.



Le langage **compte moins**



Exemple Jeff:

- Dév expérimenté
- A développé around.us
- Technos mobile & go, jamais utilisées
- A sorti l'appli hyper rapidement, depuis elle trust le haut de l'app store dans la catégorie voyage



Une bonne développeuse sait produire **efficacement** une application
à partir d'un besoin métier, en apportant sa vision technique,
gérant la complexité intrinsèque,
évitant la complexité artificielle et s'adaptant au contexte.



Si on reprend la définition que j'ai proposé en début de conf, tout reste valable avec l'arrivée des LLM. Qu'on s'en serve ou non.

Cependant il y a un point qui évolue:

Efficacement. Très clairement l'IA est en train de redéfinir les standards en matière de délais de livraison, il faut s'y préparer car même si en France il y a encore beaucoup d'organisations qui sont au stade de l'expérimentation, tout cela va s'accélérer, **encore plus avec les difficultés économiques des entreprises.**

Cependant **l'IA ne remplacera pas demain les dévs.** Par contre elle remplacera les dévs qui ne s'intéressent qu'au code mais ne font pas partie du top 5% mondial.

Conclusion



On ne sait pas de quoi demain sera fait. On est clairement au début de l'histoire, n'oubliez pas ce qui fait de vous de bons développeurs et de bonnes développeuses. Car **vos compétences ne seront pas obsolètes**, il y a un changement important qui se produit en ce moment même et c'est une formidable opportunité.

Aujourd'hui ma plus grande incertitude va concerner le coût des outils. Lorsque la bulle explosera, il est possible que le coût des outils augmente fortement. Il faudra donc probablement arbitrer ce qui doit être produit par le LLM ou non. Mais il y a fort à parier que même si les ratios vont varier, le cout d'une ligne produite à l'aide d'un LLM sera toujours fortement inférieur à celui d'une ligne produite manuellement. Ces outils sont donc là pour rester. **Le plus tôt vous les maîtriserez, le plus employable vous le serez.**

La plupart du temps, qu'est ce qui sépare un junior d'un senior ? Le temps. **Ne prenez donc pas de retard si vous ne souhaitez pas être les juniors de demain.**

Merci pour votre attention, cette conf fait partie d'une série de 4 confs dédiées à l'IA lors du Forum PHP

09 & 10 Octobre 2025
Hotel New-York - TAOM
Disneyland Paris



► event.afup.org



CRÉER UN RAG EN PHP : DÉMYSTIFIER L'IA

Clément TALLEU

Salle Olivier MANSOUR - Vendredi 09h45



09 & 10 Octobre 2025
Hotel New-York - TAOM
Disneyland Paris



► event.afup.org



CRÉER UN SERVEUR MCP AVEC SYMFONY

Edouard COURTY

Salle Cécile HAMEREL - Jeudi 10h



09 & 10 Octobre 2025
Hotel New-York - TAOM
Disneyland Paris

FORUMPHP
PARIS 2025

event.afup.org

**180 000 REQUÊTES PAR
SECONDE EXPLIQUÉ
SIMPLEMENT**

Xavier LEUNE
Salle Adrien GALLOU - Jeudi 15h40

afup

Attention ma conf sur la perf aura lieu cet après-midi.



Il sera aussi question d'IA lors de la table ronde de demain sur l'avenir de PHP

Questions ?



Open Feedback

